Navigation Recommender: Real-Time iGNSS QoS Prediction for Navigation Services

by

Duangduen Roongpiboonsopit

B.E., Chiang Mai University, 2002

M.S., University of Pittsburgh, 2006

Submitted to the Graduate Faculty of

School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Duangduen Roongpiboonsopit

It was defended on

August 15, 2011

and approved by

Dr. Prashant Krishnamurthy, Associate Professor, School of Information Sciences

Dr. Jeen-Shang Lin, Associate Professor, School of Engineering

Dr. Paul Munro, Associate Professor, School of Information Sciences

Dr. Abdelmounaam Rezgui, Associate Professor, Center for Intelligent Spatial Computing for Water/Energy Science, George Mason University

Dissertation Advisor: Dr. Hassan Karimi, Associate Professor, School of Information

Navigation Recommender: Real-Time iGNSS QoS Prediction for Navigation Services

Duangduen Roongpiboonsopit, PhD

University of Pittsburgh, 2011

Global Navigation Satellite Systems (GNSSs), especially Global Positioning System (GPS), have become commonplace in mobile devices and are the most preferred geo-positioning sensors for many location-based applications. Besides GPS, other GNSSs under development or deployment are GLONASS, Galileo, and Compass. These four GNSSs are planned to be integrated in the near future. It is anticipated that integrated GNSSs (iGNSSs) will improve the overall satellite-based geo-positioning performance. However, one major shortcoming of any GNSS and iGNSSs is Quality of Service (QoS) degradation due to signal blockage and attenuation by the surrounding environments, particularly in obstructed areas. GNSS QoS uncertainty is the root cause of positioning ambiguity, poor localization performance, application freeze, and incorrect guidance in navigation applications.

In this research, a methodology, called iGNSS QoS prediction, that can provide GNSS QoS on desired and prospective routes is developed. Six iGNSS QoS parameters suitable for navigation are defined: visibility, availability, accuracy, continuity, reliability, and flexibility. The iGNSS QoS prediction methodology, which includes a set of algorithms, encompasses four modules: segment sampling, point-based iGNSS QoS prediction, tracking-based iGNSS QoS prediction, and iGNSS QoS segmentation. Given that iGNSS QoS prediction is data- and compute-intensive and navigation applications require real-time solutions, an efficient satellite selection algorithm is developed and distributed computing platforms, mainly grids and clouds, for achieving real-time performance are explored. The proposed methodology is unique in several respects: it specifically addresses the iGNSS positioning requirements of navigation systems/services; it provides a new means for route choices and routing in navigation systems/services; it is suitable for different modes of travel such as driving and walking; it takes high-resolution 3D data into account for GNSS positioning; and it is based on efficient algorithms and can utilize high-performance and scalable computing platforms such as grids and clouds to provide real-time solutions.

A number of experiments were conducted to evaluate the developed methodology and the algorithms using real field test data (GPS coordinates). The experimental results show that the methodology can predict iGNSS QoS in various areas, especially in problematic areas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

The inspiration of this research began during the blooming of personal navigation devices for cars. My curiosities about navigation systems and Geographic Information System (GIS) in general motivated me to explore the field of Geoinformatics research with my advisor, Dr. Hassan A. Karimi, who has been working on research in navigation since the early development of navigation systems and is a pioneer in the field of Geoinformatics.

I would like to express my most sincere thanks and gratitude to Dr. Karimi for invaluable advice, patient guidance, constructive comments, and cheerful encouragement throughout the course of my research. I am also grateful to my dissertation committee members for their insightful comments. I would like to give special thanks to my former and current colleagues at the Geoinformatics Laboratory that have helped me through these difficult years. I am indebted to the Royal Thai Government who gave me an opportunity to pursue my graduation studies with full financial support.

Most importantly, none of this would have been possible without the love and patience of my family to whom I dedicate this dissertation.

1.0 INTRODUCTION

1.1 PROBLEM STATEMENT

Navigation in this dissertation is defined as the process of monitoring and guiding a vehicle from one place to another. Navigation assistance is a means by which people can be provided with navigation information according to their needs and preferences. Generally, navigation can be classified into different categories. One category, based on mode of operation, includes ground navigation, air navigation, space navigation, and marine navigation. A second category, based on navigation techniques, includes Dead Reckoning (DR), piloting, celestial navigation, and inertial navigation (Bowditch, 2002). Another category, based on navigation environment, includes outdoor navigation and indoor navigation. In the context of this dissertation, navigation is a reference to ground navigation in outdoors.

Navigation technology has experienced tremendous increase in demand over the past several years. While the rapid development of navigation devices can be attributed to the success of the Global Positioning System (GPS), the continuous development can be attributed to availability of map databases in different geographic areas, accessibility to wireless communications, and advances in mobile devices, including cell phones and smartphones. Today, navigation devices are commonplace in cars and are widely used by drivers and pedestrians.

With advancements in computer and telecommunication technologies, the development of navigation technology has evolved from *navigation systems* to *navigation services* and now to *navigation applications*. Figure 1.1 shows the relationship among these navigation systems, services, and applications. Table 1.1 summarizes the unique characteristics of navigation systems and services.

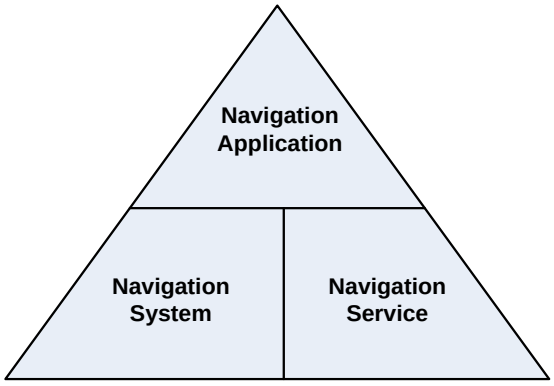


Figure 1.1. Navigation systems, services, and applications

Table 1.1. Characteristics of navigation systems and navigation services

	Navigation Systems	**Navigation Services**
Map update	Initiated and maintained by the user	Initiated and maintained by service providers
Environment	Dedicated to navigation assistance, no other systems/applications	Navigation assistance is one among other services/applications
Platform	Standalone devices installed in cars; compact, customized, portable devices	Client-server; cell phones, PDAs, and smartphones are example clients
Cost	Device purchase	Fee (e.g., per month, per usage)

Navigation systems are standalone (non-distributed) systems that can provide navigation assistance without needing and relying on external resources (e.g., external/remote computing platforms). Early navigation systems were installed onboard vehicles (in-car navigation systems offered by many automobile manufacturers). However, due to the high cost and the permanent installation of in-car navigation systems, modern navigation systems, Personal Navigation Devices (PNDs), are portable and affordable (e.g., Garmin, TomTom, Magellan).

Navigation services are client-server (distributed) systems that can provide navigation assistance by utilizing various geographically distributed resources (e.g., external/remote map

databases). The current trend in navigation services is thin navigation applications installed on mobile devices but map databases and navigation computations are preferred on remote servers. Thus, users of navigation services are not responsible for map update as required in navigation systems. In addition, other real-time information (e.g., traffic, weather) can be tightly integrated with navigation solutions in order to improve user's real-time experience. Another feature offering a new way of utilizing navigation assistance is flexible fee (e.g., per months, per usage). Examples of navigation service providers are Verizon Navigator (Dolan, 2007), Google Maps Navigation (Official Google Blog, 2009), and Nokia Navigation (Dennehy, 2010).

Navigation applications are programs/software installed on mobile devices that can provide navigation assistance according to user's requirements or preferences such as different modes of operations (i.e., driving, walking, riding bikes or wheelchairs), routing criteria (i.e., shortest route, fastest route, personalized routes, GNSS QoS-based routes). They can be implemented as standalone applications in navigation systems or as clients requesting navigation information through navigation services. Navigation information presented in each application can be customized to fulfill the need of its target users. Examples of navigation applications on smartphones are MobileNavigator, TomTom for iPhone, Magellan RoadMate, and Garmin Streetpilot.

Throughout this dissertation, navigation systems, navigation services, and navigation applications are used interchangeably.

Modern navigation technology composes of five modules: map database, geocoding, geo-positioning, map matching, and routing and direction (Skog and Händel, 2009, Zhao, 1997). Each navigation module is susceptible to errors which contribute to the overall uncertainty in a navigation system/service (Karimi et al., 2011a).

- *Map Database*. A map database contains the geometry, topology, and attributes of a navigable network (e.g., road network, sidewalk network). Road network data errors come from multiple sources during the map generation process. The quality of map databases is crucial to perform navigation tasks (i.e., geocoding, map matching, routing).
- *Geocoding*. Geocoding is the process of converting a given place name to geographic coordinates on a map. Its errors are mainly associated with choices of techniques (street geocoding or rooftop geocoding), algorithms and reference

databases (Roongpiboonsopit and Karimi, 2010a, b). Inaccurate geocoded destinations may result in non-optimal routes.

- *Geo-positioning*. Geo-positioning is the process of estimating best location updates of a vehicle/user in real time. A position is estimated from measurements (or observations) of a sensor(s) and is subject to noises and various sources of errors. Since positioning is a real-time process, navigation systems/services are unaware of positioning quality until the user is at the location.
- *Map Matching*. Map matching is the process of determining current vehicle's location on the road segment utilizing the geographic coordinates obtained by the geo-positioning module. Map matching may inaccurately match positions depending on the quality of geo-positioning, the quality of map database, and the assumptions in the map matching algorithm. Karimi et al. (2006) developed a methodology to predict quality of map-matched points.
- *Routing and direction.* Routing and direction computes user-preferred routes and step-by-step instructions on how to travel on routes. The major sources of errors in computed routes are the quality of map databases and the nature of the heuristic algorithms.

As geo-positioning is a key module directly affecting real-time navigation performance, this dissertation investigates the quality of geo-positioning in order to provide navigation recommendations that maximize positioning quality. Today, many geo-positioning sensors are available including Global Navigation Satellite Systems (GNSSs), Inertial Navigation Systems (INS), DR, cellular network, Wireless Local Area Network (WLAN), Ultra Wide-Band (UWB), Radio Frequency Identification (RFID), and Bluetooth (Grejner-Brzezinska, 2004, Hightower and Borriello, 2001, Porretta et al., 2008, Retscher and Kealy, 2006, Rizos, 2005). Among them, GNSS is commonly used due to its high positional accuracy (< 10 m), global and continuous service, and low cost to the end user. In addition, GNSS-enabled mobile devices, such as smartphones, are becoming commonplace; the Allied Business Intelligence (ABI) Research has forecasted that nine of every ten smartphones will be GNSS-enabled in 2014 compared to one in three in 2008 (Zahradnik, 2009). It is most likely that navigation services, as well as other Location-Based Services (LBSs), will be developed for GNSS-enabled handsets. With

continuous improvements of positioning performance and high availability on mobile devices, GNSS has been the first sensor of choice for navigation systems/services.

GNSS is a satellite-based, all-weather, continuous, global radionavigation, and time-transfer system that allows the receiver to determine its current position. The first well-known and currently active GNSS is the US GPS. In the near future, three other GNSSs, i.e., the Russian GLONASS, the European Galileo, and the Chinese Compass, will become fully operational as alternatives to GPS. With multiple GNSSs available, researchers have explored the use of integrated GNSSs (iGNSS) which, among other benefits, will have improved availability, reliability and accuracy than using a single GNSS (Constantinescu and Landry Jr., 2005, O'Keefe et al., 2002, Ochieng et al., 2001, Wu et al., 2003). To stimulate the emergence of iGNSS, several GNSS chipset and platform manufacturers are developing chipsets capable of multiple constellations (i.e., GPS/GLONASS and GPS/GLONASS/Galileo) for both survey-grade receivers and mobile phones (Mobile Marketing, 2011, Sage, 2009).

Despite the expected improvement, GNSS, and iGNSS, is subject to two types of errors: systematic and random. Systematic errors present biases to ranging measurements and position estimation, including satellite orbital errors, satellite and receiver clock errors, reference station position errors, satellite geometry and signal propagation errors. Other errors, such as receiver noise, are considered random. Most of the biases at the global or regional scale (i.e., satellite orbital errors, satellite and receiver clock errors, ionospheric and tropospheric effects) can be modeled mathematically and accounted for in order to remove or reduce their effects (Hofmann-Wellenhof et al., 2008, Kaplan and Hegarty, 2006). However, biases at the local scale (i.e., multipath and attenuation) still remains uncorrected because it is highly dependent on the local environment surrounding the receiver. Transmitted signals from GNSS satellites are subject to signal blockage, attenuation, reflection, and diffraction by nearby objects before arrival at the receiver. As a result, GNSS receivers may provide poor positions with low accuracy (> 50 m) or no solution. The local scale biases are major drawbacks of individual GNSS as well as iGNSS.

The main reason why the local scale biases remain unsolved is that the GNSS receiver does not have any information about nearby environmental structures. Thus, the level of measurement errors due to signal attenuation and multipath effect is not known by the receiver. This prevents the receiver from totally eliminating multipath errors or accurately predicting positioning quality itself. In addition, it is impossible to predict positioning quality of the

expected locations in advance since the receiver does not have knowledge of whereabouts the user is expected to be. At best, the receiver can approximately compute its positioning quality in real time from received signal strengths, estimated errors, and satellite geometry. This real-time positioning quality estimation is not highly reliable due to the existence of several hidden errors (e.g., signal path delays).

As a consequence, GNSS-based applications (navigation and other LBSs) relying solely on real-time positioning quality estimated by the GNSS receiver are also subject to real-time performance degradation. Operating under obstructed areas, such as urban canyons, canopies, or indoor environments, the GNSS receiver usually experiences two major problems: GNSS outages and poor GNSS positions. GNSS outage can occur any time without any warning which cause unexpected application freeze. Poor GNSS positions can cause application malfunctions, such as giving incorrect route directions or frequently re-computing new routes. Poor GNSS solutions (at least 100 m) can have dire consequences in critical applications, for example, a fatal accident of two sailors (AAP, 2010).

Among GNSS-based applications suffering from real-time positioning uncertainty, navigation applications have an advantage that user's future location can be predicted through computed routes. Thus, it is possible for navigation applications to reduce or avoid the effect of poor GNSS positioning quality, or GNSS Quality of Service (QoS), if GNSS QoS along the route is known in advance. However, such GNSS QoS prediction is currently not available. Current navigation applications cannot provide alternative navigation solutions or routing decisions based on GNSS QoS.

While augmenting GNSS with other geo-positioning sensors (e.g., DR for in-car navigation systems) is possible to improve positioning solutions in navigation systems, employing those sensors on mobile devices, where resources are limited, may not be practical. There are several existing and emerging sensors, such as Wi-Fi Positioning System (WPS), cellular-based positioning system, and Vision-based Positioning System (VPS), possible to augment GNSS. However, the main shortcoming of these other sensors is position availability is limited within certain areas. For example, WPS may not be available in an area with poor WLAN coverage. VPS may be possible only in the areas that have reference images stored in a remote server (Katsura et al., 2003). Although using all available positioning sensors simultaneously is a possible approach, it is not practical, especially on a mobile handset, since

utilizing all geo-positioning sensors concurrently requires significant amount of computing power from the mobile device. With provision of GNSS QoS prediction, it is expected that navigation applications can prepare for an appropriate GNSS augmentation before experiencing poor or no GNSS solutions.

As mobile handsets become commonplace for navigation applications and many new LBSs that require continuous tracking, managing power consumption will be a challenging task. Continuous computation by GNSS sensor and other augmented sensors significantly shortens battery life. It is envisioned that, with GNSS QoS prediction, navigation applications can plan for activating appropriate geo-positioning sensors in order to optimize power consumption while maximizing navigation performances. For example, activate GNSS when GNSS QoS is high and deactivate GNSS and activate the augmentation module when GNSS QoS is low.

Predicting GNSS QoS is the prime motivation for this research. We define "*iGNSS QoS prediction*" as the level of positioning QoS prediction by two or more GNSSs at a given location and time. Of the applications that could benefit from iGNSS QoS prediction, navigation applications are particularly important in that: (a) they require high-quality positioning solutions to be of practical use and (b) performance of a navigation application is crucial to the user who has to make navigation decisions in a short time span.

1.2 BENEFITS OF IGNSS QOS PREDICTION

Potential benefits of predicting iGNSS QoS in navigation systems/services are:

- Providing awareness about the areas with low or high positioning quality
- Planning routes based on GNSS QoS
- Planning for geo-positioning augmentation using other geo-positioning sensors
- Optimizing energy consumption

1.3 CHALLENGES OF IGNSS QOS PREDICTION

Developing iGNSS QoS prediction for navigation systems/services involves several challenges.

First, predicting iGNSS QoS requires spatio-temporal modeling as iGNSS QoS varies across space and time. GNSS satellite dynamism and navigation user dynamism are causes of spatio-temporal variations in GNSS QoS especially in obstructed areas. GNSS satellites are orbiting around the Earth in different orbital planes with different motion from the Earth's rotation; thus, at any location on the Earth, a different set of satellites can be observed at different times. In addition, users of navigation systems/services move from one place to another where surrounding environments may dramatically change (e.g., from open sky to completely blocked condition).

Second, accurate iGNSS QoS prediction is data- and compute-intensive due to the use of realistic representation of surrounding environments. Although several simulation tools for predicting GNSS performances have been reported in the literature (see Section 2.4 for details), they are not widely used and not applicable for iGNSS QoS prediction due to some major shortcomings. First, there is a lack of available accurate 3D data that represents the surrounding environment of the predicting location, e.g., terrain and buildings. Second, most simulation tools are tightly coupled with proprietary functions or data formats. Third, computation using 3D data can be intensive and time-consuming.

Third, as navigation is the focus of this research, iGNSS QoS prediction must be computed in a short time interval (i.e., real-time). The predictive information must be available to the navigation application before the user travels in the areas; otherwise, the information is obsolete. With such a real-time constraint, the application also expects precise iGNSS QoS prediction. Currently, there is a void in the GNSS literature on time performances of existing GNSS simulations using 3D data for predicting QoS of positioning solutions in a large area.

Last, using iGNSS increases complexity and workload of prediction due to the increased number of GNSS satellites, about 100 compared to 24 of GPS only. iGNSS QoS simulation must consider all possible satellites and evaluate all possible solutions by different combinations of satellites. Existing GNSS simulation tools do not provide suggestion of when and where to use which mode of operations, single or multiple GNSSs, and what combination of satellite constellations would be optimal for the receiver.

With an understanding of these challenges, novel methodologies and algorithms for predicting iGNSS QoS that: (a) are suitable for navigation applications; (b) can respond in real time; and (c) are able to provide navigation recommendations based on iGNSS QoS prediction, are needed.

1.4 UNIQUE CHARACTERISTICS OF IGNSS QOS PREDICTION

Comparing to other existing GNSS performance simulation tools, iGNSS QoS prediction has unique characteristic as follows:

- It addresses positioning requirements of navigation systems/services
- It provides a new means of route choices and routing
- It is suitable for different modes of travel
- It takes high-resolution 3D data into account
- It is based on efficient algorithms and can utilize distributed computing to provide real-time response

1.5 GOAL AND OBJECTIVES

The ultimate goal of this research is to provide iGNSS QoS prediction on routes and to recommend alternate navigation solutions based on the predicted iGNSS QoS. To achieve this goal, the following research objectives are set forth in this dissertation:

- To identify and formulate the iGNSS QoS parameters that are important to navigation applications
- To develop a new methodology to predict iGNSS QoS for navigation applications
- To evaluate the developed methodology by comparing the prediction results with actual GNSS data
- To develop an efficient algorithm and explore existing distributed computing platforms (i.e., grid and cloud) for real-time iGNSS QoS prediction

- To demonstrate navigation recommendations based on iGNSS QoS prediction

1.6 CONTRIBUTIONS

This research contributes the following:

- A set of iGNSS QoS parameters for navigation applications, which are visibility, availability, accuracy, continuity, reliability, and flexibility
- A new methodology for iGNSS QoS prediction. The followings are new algorithms developed for this methodology:
    - A set of algorithms for visibility calculation of direct, diffracted, and reflected Line-of-Sights (LOSs) based on high-resolution Triangulated Irregular Networks (TINs)
    - A new algorithm for determining the point sequence of iGNSS QoS chunks, called Sequence Merging Algorithm (SMA)
- An efficient algorithm for iGNSS satellite selection, called Multi-Constellation Satellite Selection Algorithm (MCSSA)
- Feasibility studies of using high-performance and scalable platforms, i.e., grids and clouds, to address the real-time constraint in iGNSS QoS prediction
- Provisions of maps with iGNSS QoS prediction for route planning
- Optimal iGNSS QoS-based routing in navigation systems/services

1.7 ORGANIZATION

This dissertation is organized into eight chapters. Chapter 1 states the motivation of this research, its goal and objectives, and its contributions.

Chapter 2 provides a background on GNSS including current and future GNSSs, the satellite-based positioning method, sources of errors, and measurement metrics of GNSS performances. It also reviews the related literature on GNSS performance prediction.

Chapter 3 defines the six parameters of iGNSS QoS necessary for navigation systems/services and gives an overview of the methodology of iGNSS QoS prediction, which encompasses four modules: segment sampling, point-based iGNSS QoS prediction, tracking-based iGNSS QoS prediction, and iGNSS QoS segmentation.

Chapter 4 describes point-based iGNSS QoS prediction, which composes of a GNSS simulation and signal propagation models. A set of algorithms are developed for determining visibility of different LOS types using TINs constructed from Light Detection and Ranging (LiDAR) point cloud. Three QoS parameters (i.e., visibility, availability, and accuracy) are computed for a given location and time. For the accuracy parameter, a fuzzy logic-based model is developed by combining several factors impacting positional accuracy.

Chapter 5 describes the details of segment sampling, tracking-based iGNSS QoS prediction, and iGNSS QoS segmentation that are needed for providing predictive information suitable for navigation systems/services. Systematic sampling with a fixed-distance interval is employed to select points on a route segment. The tracking-based prediction module computes four QoS parameters (i.e., average availability, average accuracy, continuity, and reliability) for each route segment. iGNSS QoS segmentation represents areas of a segment with similar QoS as chunks.

Chapter 6 describes two approaches for enhancing computation time of iGNSS QoS prediction. One involves reducing the number of iGNSS satellites that need to be included in point-based iGNSS QoS prediction. A satellite selection algorithm is developed for this purpose. Another is to explore performances and limitations of grid and cloud computing platforms for addressing the real-time constraint in iGNSS QoS prediction.

Chapter 7 discusses the benefits of the outcome of the research by demonstrating two examples: provision of maps with static iGNSS QoS prediction for route planning and provision of optimal routing based on iGNSS QoS. Sample applications that can take advantages from iGNSS QoS prediction are discussed.

Chapter 8 summarizes the research and its contributions, discusses conclusions, and provides recommendations for future research.

2.0 BACKGROUND AND RELATED WORK

This chapter provides background on GNSSs and related work on GNSS performance prediction. We first describe basic components of GNSSs, summarize the current and planned GNSSs, and briefly discuss the satellite-based positioning principle and GNSS budget errors. These are followed by an overview of the metrics commonly used for measuring GNSS performances. The Required Navigation Performance (RNP) is then discussed. This chapter ends with a review of the literature on existing GNSS performance simulation tools.

2.1 GLOBAL NAVIGATION SATELLITE SYSTEM

GNSS is the generic term referring to a satellite-based positioning network. The satellites broadcast high frequency radio signals containing transmitted time and navigation code, allowing the GNSS receiver to determine its location anywhere and anytime. This passive process of determining a position at the receiver side allows GNSS to serve an unlimited number of users over the world. GNSS is comprised of three main segments: space segment, control segment, and user segment (Drane and Rizos, 1998, Hofmann-Wellenhof et al., 2008, Parkinson and Spilker, 1996).

Space segment consists of constellation of satellites, including active and spare satellites. Satellites are responsible for receiving and storing data from the control segment, maintaining accurate time using several onboard atomic clocks, and transmitting information on carrier signals to users on the designated frequency bands.

Control segment consists of ground facilities responsible for monitoring satellite health and the downlink signals, commanding and controlling the satellites, collecting atmospheric data,

updating navigation messages, and computing satellite orbit and clock data. Several ground stations are usually located at different positions across the globe.

User segment is the receiving equipment that processes signals transmitted from the satellites to compute position, velocity, and time solutions.

2.1.1 Present and Future Systems

Currently, four systems are considered as core GNSSs: GPS, GLONASS, Galileo, and Compass. A brief overview of each GNSS is presented below. For a detailed description of each system, refer to Hofmann-Wellenhof et al. (2008) and Kaplan and Hegarty (2006). Table 2.1 summarizes system segments and signal characteristics of each GNSS, adapted from Hein et al. (2007), Hofmann-Wellenhof et al. (2008), Kaplan and Hegarty (2006), and Kleusberg (1990).

GPS

The NAVigation System with Timing And Ranging (NAVSTAR) or GPS was initiated by the US Department of Defense (DoD). The project started in the early 1970s, reached Initial Operational Capability (IOC) in 1993 and declared the Full Operational Capability (FOC) in April 1995. Later in 1996, the US recognized the importance of GPS to civilian users by declaring a dual-use system of GPS. Certain signal capabilities are reserved for US military applications while the civilian signals are accessible to worldwide users.

GPS provides two main services: Standard Positioning Service (SPS) and Precise Positioning Service (PPS). SPS is a positioning and timing service available to all users on a continuous, worldwide basis with no direct change. SPS currently provides GPS L1 frequency which contains a coarse acquisition (C/A) code and a navigation data message. It is specified to provide global average of $\leq$ 9 m horizontal accuracy and $\leq$ 15 m vertical accuracy (95% probability) (US DoD, 2008). PPS is a highly accurate military positioning, velocity and timing service (better than SPS) for military and government agency users and the civilian users can access PPS with US DoD approval. PPS currently provides navigation data on GPS L1 and L2 frequencies encrypted with P(Y) code that is only available to military equipment with a proper decryption key.

GPS constellation consists of 24 operational satellites in six Medium Earth Orbit (MEO) planes with four active satellites per plane plus several active spare satellites for replenishment. The orbital planes are evenly spaced around the equator at a 60° separation with an inclination angle of 55° referencing to the equatorial plane. Satellite orbits are non-geostationary at an altitude of about 20,200 km above the Earth's surface. Orbital period is one-half of a sidereal day, about 11 hours 58 minutes. This particular design allows most locations to observe three or more GPS satellites.

GLONASS

GLObal'naya NAvigatsionnaya Sputnikovaya Sistema, or GLObal NAvigation Satellite System (GLONASS), was developed by the former Soviet Union and currently operated by the Russia Space Forces. It was initiated in the mid-1970s as a military system and officially declared operational in September 1993. In March 1995, the Russian government declared an open service of GLONASS for civilian use. However, FOC was not declared until January 1996 with the nominal 24-satellite constellation. Due to financial problems, the number of satellites declined soon after the declaration of FOC. GLONASS reached its minimum 6-8 satellites in 2001. The Russian government decided to revitalize the system with a guarantee for funding from 2002 to 2011. As of 2011, there are 23 satellites in operation, 1 satellite in commissioning phase, and 3 satellites in maintenance[1].

Similar to GPS, GLONASS provides two services: one for civilian use and one for military use. The performances of these services are not officially revealed. Based on measurements from four satellites, Space Today Online[2] states that the accuracy of civilian service is within 55 m horizontal accuracy and 70 m vertical accuracy. For the military service, it is expected to provide highly accurate solutions at a similar level to the PPS of GPS.

GLONASS constellation was designed with 24 satellites in three orbital planes with seven active satellites and one active spare per plane. The planes are separated by 120° with an inclination angle of 64.8° to the equator. Each satellite is orbiting at an altitude of 19,100 km above the ground and completes the orbit in approximately 11 hours 15 minutes. This

[1] http://www.glonass-ianc.rsa.ru/en/
[2] http://spacetoday.org/Satellites/GLONASS.html

constellation assures that at least five satellites are simultaneously visible by 99% of the Earth's surface (Habrich, 1999).

Galileo

Galileo is a GNSS under development funded by the European Union (EU) and European Space Agency (ESA) since March 2002. The original plan required to launch 30 satellites through 2006-2010, however due to financial problems the plan has been shifted. It is anticipated that FOC of Galileo will be around 2020. Unlike GPS and GLONASS, Galileo is under civilian control and intended primarily for civilian use. The full precision will be available to all users.

Five services are defined when Galileo is fully operational: Open Service (OS), Commercial Service (CS), Safety of Life (SoL), Public Regulated Service (PRS), and the support for a Search and Rescue (SAR) service. OS is the only free accessible service to all users with 15-24 m horizontal accuracy and 35 m vertical accuracy (95% probability) with the use of single frequency, and with 4 m horizontal accuracy and 8 m vertical accuracy (95% probability) with the use of dual frequency. OS provides only basic navigation information while other information, such as integrity, continuity, or service guarantees, is preserved for specific groups of users. CS offers the same level of accuracy as OS and provides service guarantees. The access to CS data message will be encrypted and controlled by the Galileo system administrators. SoL relies on the same signals as OS but adds integrity and continuity information and will be primarily used by aviation applications. PRS will be offered only to government-authorized users and aimed to provide a continuous, robust, and encrypted signal that will be usable in situations of crises while other services are unusable. Its performance is envisioned to be comparable to OS while providing integrity information comparable to SoL. Galileo satellites are designed to support worldwide humanitarian SAR operations by providing the emergency beacons to the SAR control centers.

The fully deployed Galileo system will consist of 27 operational satellites plus 3 non-active spares in three MEO planes. The planes are separated by 120° with an inclination angle of 56° with reference to the equatorial plane. The satellites are orbiting at 29,601 km above the ground with a revolution period of 14 hours 4.75 minutes. With FOC, the design of Galileo constellation will provide good coverage even at latitudes above 75° north and south.

Compass

China's navigation system, called Beidou, was developed in 1993 with a two-step approach. The first step, Beidou-1, is to develop a regional augmentation system. The current Beidou-1 system, completed in 2007, consists of 4 satellites. The second step, Beidou-2, is to add more satellites to Beidou-1 for the purpose of higher performances. During the design of the second step, the concept has been expanded to build up a global system, called Compass.

Compass will be a dual-use system with civilian open service and authorized service. The open service is designed to provide positional accuracy of 10 m. No detailed performance has been revealed for the authorized service.

Compass constellation will consist of 27 MEO satellites, 5 satellites in Geostationary Earth Orbit (GEO), and 3 satellites in geosynchronous orbit. The 24 MEO satellites will be evenly spaced with 45° separated in three orbital planes with an inclination angle of 55°. The additional three MEO satellites are presumably spare ones. All MEO satellites will have an average orbital altitude of 21,500 km above the Earth. The three geosynchronous orbit satellites have an altitude at 35,785 km above the Earth with an inclination angle of 55°. They will position in the three orbital planes with fixed positions.

Table 2.1. Summary of present and future GNSS (after Hein et al. (2007), Hofmann-Wellenhof et al. (2008), Kaplan and Hegarty (2006), and Kleusberg (1990))

Characteristic	GPS	GLONASS	Galileo	Compass
Space segment				
• No. of operational SVs	24	24	27	24
• No. of orbital planes	6	3	3	3
• SVs per orbital plane	4 + 1 spare	8 + 1 spare	9 + 1	6 + 1 spare
• Angles between planes	60°	120°	40°	
• Altitude (km)	20,200	19,100	23,222	21,150
• Inclination angle	55°	64.8°	56°	55°
• Orbital period	11 h 57.96 min	11 h 15.73 min	14 h 4.75 min	12 h 36 min
• Ground track repeat period	~1 sidereal day	~8 sidereal days	~10 sidereal days	~7 sidereal days
• Ground track repeat orbits	2	17	17	-
Control segment	· 1 Master control station (MCS) · 5 Monitor stations	· 1 System control center (SCC) · 1 Central Synchronizer	· 5 Telemetry, tracking and command stations (TT&C) · 9 (Mission data) uplink stations (ULS)	N/A

Characteristic	GPS	GLONASS	Galileo	Compass
	· 3 Ground stations	· 1 Phase control system · 3 Command and tracking stations · 2 Laser tracing stations · 2 Navigation field control equipments	· 40 Galileo sensor stations (GSS)	
Signal Characteristic				
• Signal separation	CDMA	FDMA	CDMA	CDMA
• Number of frequencies	3 – L1, L2, L5	One per two antipodal SV	3(4) – E1, E6, E5 (E5a, E5b)	4 – E1, E2, E5B, E6
• Carrier Frequency (MHz)	L1: 1,575.420 L2: 1,227.600 L5: 1,176.450	L1:(1,602+k(9/16) L2:(1,246+k(7.16) k = channel number	E1: 1,575.420 E6: 1,278.750 E5: 1,191.795	N/A
• Number of ranging codes	11	6	10	N/A
• Code clock rate (MHz)	C/A: 1.023 P: 10.23	C/A: 0.511 P: 5.11	E1: 1.023 E6: 5.115 E5: 10.23	N/A
• Code length (chips)	C/A: 1,023 P: 6.187104×10^{12}	C/A: 511 P: 5.11×10^{6}	E1: 4,093 E6: 5,115 E5: 10,230	N/A
• Integrity transmission	No (GPS III:yes)	No (GLONASS-K: yes)	yes	N/A
C/A-code Navigation Message				
• Superframe duration (min)	12.5	2.5	N/A	N/A
• Superframe capacity (bits)	37,500	7,500	N/A	N/A
• Superframe reserve capacity (bits)	~2,750	~620	N/A	N/A
• Word duration (seconds)	0.6	2	N/A	N/A
• Word capacity (bits)	30	100	N/A	N/A
• Number of words within a frame	50	15	N/A	N/A
• Technique for specifying satellite ephemeris	Keplerian elements, correction coefficients	Geocentric Cartesian coordinates and their derivatives	Keplerian elements, correction coefficients	Keplerian elements, correction coefficients
• Time reference	GPS Time, UTC (USNO)	GLONASS Time, UTC (SU)	Galileo time	N/A
• Position reference (geodetic datum)	WGS-84	PE-90	GTRF	N/A

2.1.2 Principle of Satellite-Based Positioning

Despite several independent GNSSs in operation, they all operate based on the fundamental principle of Time of Arrival (TOA) ranging. GNSS satellites transmit high frequency signals containing time and ranging codes containing navigation messages to the Earth. The basic measurement made by a GNSS receiver is the time each signal took from leaving a GNSS satellite to arriving at the receiver. Because the signal travels at the speed of light, this time interval can be converted to a distance, called a pseudorange. Once the distance to available satellites has been determined, the position can be calculated by using trilateration techniques. To determine a 2D position, at least three satellites are required while at least four satellites are required for computing a 3D position. For further details on position calculation refer to Hofmann-Wellenhof et al. (2001), Kaplan and Hegarty (2006), Langley (1991), and Parkinson and Spilker (1996).

To measure travel time of satellite signals, there are two types of measurements in GNSS receivers: range and carrier-phase observations (Drane and Rizos, 1998). Range observations are based on signal codes (e.g., GPS transmits the C/A-code for civilian users and the P-code for military users), sometimes referred to as code range or code phase. A specific code, e.g., GPS has Pseudorandom Noise (PRN) code, marks accurate time on each signal when it is transmitted. The receiver determines the travel time of an incoming signal from a satellite by comparing the receiver-generated codes with an identical code in the signal. The amount of codes that the receiver slides later in time to sync up with the signal codes is equal to the signal's travel time. However, the main problem of range observations is that the size of signal codes is rather wide which can cause errors in the measurement. For example, the C/A-code has a chip rate of 1.023 Mbps, which corresponds to a resolution of about 300 m per chip at a speed of light, and the P-code has a chip rate of 10.23 Mbps, which yields a higher resolution of about 30m. Given that the incoming and receiver-generated codes can be generally aligned within about 1-2% of the chiplength, the measurement precision of C/A-code ranges about 3-5 m and for P-code ranges about 0.3-0.5 m.

Carrier-phase observations are based on the carrier phase of the signal. The receiver counts the exact number of carrier cycles that the signal takes from the satellite to the receiver. Thus, this method does not require processing the information in the signal. The carrier

frequencies are much higher than the signal codes, which give more precise measurements. For example, the L1 and L2 carrier waves of GPS have relatively short wavelengths about 19 cm and 24 cm, respectively, comparing to the C/A- and P-code. Assuming a measurement resolution of 1-2% of the wavelength, the measurement precision of carrier-phase ranging is at millimeter level comparing to a few meters of code ranging. However, the problem of carrier-phase observations is that the carrier waves are uniform (i.e., every pulse cycle looks like every other) causing ambiguity to discriminate between one wavelength from another. Unlike the code ranging that can determine the time of transmission from the modulated code (e.g., PRN), carrier-phase observations do have an exact reference causing difficulty in counting wave cycles. Carrier-phase ranging is nevertheless the basis for high-precision positioning in survey-grade receivers that combine code and carrier-phase observations for high accurate measurements.

In this dissertation, we focus on positioning with code range observations since it is supported by common GNSS receivers used in navigation systems/services.

2.1.3 Error Budget of GNSS Observables

Positioning quality of GNSS receivers, i.e., code range or carrier phase, is dependent on a complicated interaction of various factors. In general, GNSS positioning quality mainly depends on the level of measurement errors and satellite-receiver geometry. Some of these measurement errors can be reduced or eliminated by GNSS augmentation techniques (e.g., differential technique). Poor satellite geometry, on the other hand, cannot be corrected. Figure 2.1 summarizes the major sources of GNSS errors. Details of each error are described as follows.

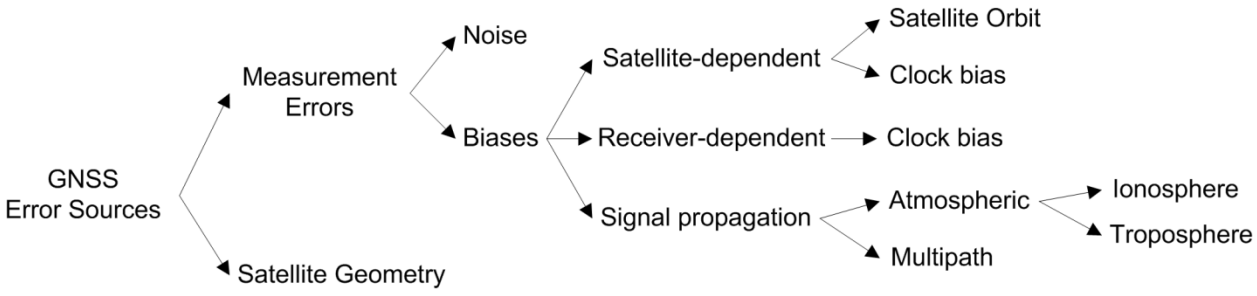


Figure 2.1. GNSS error sources

2.1.3.1 Measurement Errors

Similar to other geo-positioning sensors, GNSS is subject to at least two types of measurement errors: random errors and biases (Drane and Rizos, 1998). Random errors occur due to stochastic noise in the measurement process, which are assumed to be Gaussian. The random errors can cause different position values in each time epoch when measuring at a fixed location.

Biases, or systematic errors, occur as non-zero outputs for zero inputs due to offsets in the measurement process. These offsets can cause constant errors over a short time period but can slowly grow with time. GNSS measurement biases can be categorized into three groups: satellite-dependent, receiver-dependent, and signal propagation. Satellite-dependent biases consist of orbital bias and satellite clock bias. Satellite orbital bias occurs when there is discrepancy between the true position of the satellites and the predicted position in ephemerides data prepared by the master control station. Satellite clock error is another unavoidable bias that changes with time but can be eliminated by the receiver using correction models. For example, the typical stand-alone GPS receivers can correct satellite clock errors by using the broadcasted clock error model defined as the three polynomial coefficients (a_0, a_1, a_2) in the navigation message. Receiver-dependent bias is clock error due to the use of inexpensive quartz crystal oscillator. Receiver clock error is the difference between the receiver clock time and the GNSS time.

Signal propagation biases occur due to atmospheric and multipath effects. Ionosphere and troposphere cause slower velocity of propagating signals comparing to velocity in the outer space. The reduced speed of propagation results in a longer measurement range. The ionospheric band of the atmosphere (a height of about 50 km to 1,000 km) contains a large number of electrons and positive charged ions released from the gas molecules being ionized by the solar radiation. Due to signal dispersion on these ions, ionospheric delay can cause bias on the pseudorange from 50 m (at the zenith) to 150 m (at the horizon). The receiver can mostly eliminate the ionospheric bias using a function of signal frequencies modeled for standard conditions of the ionosphere (Hofmann-Wellenhof et al., 2001). However, unforeseen conditions (e.g., strong solar winds) are impossible to be corrected by civil receivers. Troposphere, the lowest layer of the atmosphere (a height of about 7 km to 20 km), contains water vapor and dust particles. Tropospheric refraction delay is caused by different concentrations of water vapor

(different weather conditions), which can result in bias on the pseudorange from 2 m (at the zenith) to 20 m (at 10 degrees elevation angle) (Brunner and Welsch, 1993). Several models of tropospheric effect are available, such as Hopfield model and Saastamoinen model (Hofmann-Wellenhof et al., 2008). However, only about 90% of this bias can be removed and the remainder is difficult to model due to water vapor.

Multipath effect occurs whenever the signal takes an indirect path, reflecting off nearby objects or diffracting from the edge of nearby objects, rather than propagating over a direct path from satellite to receiver. For code range observations, the delay path can cause positioning errors about 10-20 m and may increase to 100 m in severe areas. Signals from low satellite elevations are more susceptible to multipath effect than those in high elevations due to possible geometry for reflection. Various methods have been developed to reduce the multipath effect, which can be classified as antenna-based mitigation, improved receiver technology, and signal and data processing (Ray et al., 1999). These methods can cope with multipath rather well for stationary mode of data collection. However, the multipath effect is more challenging in dynamic mode of moving objects due to varying surroundings from place to place. The multipath effect causes the most significant bias to the final position solution in the most common mode of use, i.e., moving objects.

To combine contributions of different sources of measurement errors described earlier, a parameter, called the User Equivalent Range Error (UERE), is calculated. Assuming that individual errors are uncorrelated, UERE associated with a satellite is computed as a square root of the summed squares of the individual errors. Table 2.2, adapted from Hofmann-Wellenhof et al. (2008) and Wormley (2009), summarizes the six measurement errors for L1 C/A code and the computed UERE. The UERE of the typical stand-alone GPS receiver is presented in the "Total GPS" column, while the UERE of the differential GPS is presented in the "DGPS" column. Differential GPS (DGPS) is the most widely used technique to overcome most errors experienced in the stand-alone GPS. However, the multipath and the receiver errors remain (as high as 81.3% of the total errors) due to the great variations of environments and types of receivers, which are difficult to model.

Table 2.2. Summary of measurement errors and UERE for L1 C/A code (after Hofmann-Wellenhof et al. (2008) and Wormley (2009))

Error source	1σ (m)			1σ (m)	% contributed to UERE	
	Bias	Random	Total GPS	DGPS	Total GPS	DGPS
Ephemerides data	2.1	0.0	2.1	0.0	16.0%	0.0%
Satellite clock	2.0	0.7	2.1	0.0	16.0%	0.0%
Ionosphere	4.0	0.5	4.0	0.4	58.1%	6.6%
Troposphere	0.5	0.5	0.7	0.2	1.8%	1.7%
Multipath	1.0	1.0	1.4	1.4	7.1%	81.3%
Receiver measurement	0.5	0.2	0.5	0.5	0.9%	10.4%
UERE, RMS	5.1	1.4	5.3	1.6	100.0%	100%

2.1.3.2 Satellite-Receiver Geometry

Satellite geometry describes the geometric relationship of the visible satellites' position and the receiver's position. Figure 2.2 illustrates the geometrical effect on positioning uncertainty with the three-satellite arrangements for calculating a 2D position, assuming the satellites have the same range of errors. A good geometry, Figure 2.2(a), refers to the case that the satellites are spatially scattered evenly over the sky view of the receiver, where the uncertainty region of the estimated position is small resulting in an accurate position. A poor geometry, Figure 2.2(c), refers to the case that the satellites are in close proximity to each other or arranged nearly to a straight line, where the uncertainty region is large resulting in a poor position solution.

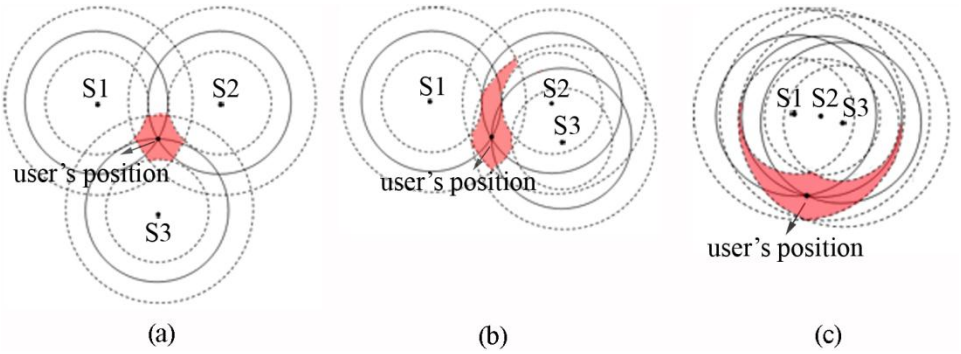


Figure 2.2. Satellite-receiver geometry effect

The parameter usually used to determine the effect of satellite geometries is Dilution of Precision (DOP) (Langley, 1999, Massatt and Rudnick, 1991, Yarlagadda et al., 2000), which is a number greater than one. Based on different components used in DOP calculations, several definitions of DOP are possible. The most common DOPs for ground navigation systems are positional DOP (PDOP for 3D coordinates) and horizontal DOP (HDOP for 2D coordinates).

DOPs are derived from the design matrix **A** in Equation 2.1, where the first three columns are the components of the unit vectors, pointing from the observing site to the satellites, and the last column represents time bias of the measurements. The cofactor matrix $\mathbf{Q_x}$ is then calculated, using Equation 2.2, in order to calculate different DOPs. Table 2.3 summarizes the definition of five DOPs.

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & z_1 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & -1 \end{bmatrix} \tag{2.1}$$

$$\mathbf{Q_x} = [\mathbf{A^T A}]^{-1} = \begin{bmatrix} q_{xx} & q_{yx} & q_{zx} & q_{tx} \\ q_{xy} & q_{yy} & q_{zy} & q_{ty} \\ q_{xz} & q_{yz} & q_{zz} & q_{tz} \\ q_{xt} & q_{yt} & q_{zt} & q_{tt} \end{bmatrix} \tag{2.2}$$

Table 2.3. DOP definitions

DOPs	Calculations	Definitions
GDOP	$\sqrt{q_{xx} + q_{yy} + q_{zz} + q_{tt}}$	Geometric Dilution of Precision
PDOP	$\sqrt{q_{xx} + q_{yy} + q_{zz}}$	Position Dilution of Precision
TDOP	$\sqrt{q_{tt}}$	Time Dilution of Precision
HDOP	$\sqrt{q_{xx} + q_{yy}}$	Horizontal Dilution of Precision
VDOP	$\sqrt{q_{zz}}$	Vertical Dilution of Precision

In general, the accuracy of a stand-alone GNSS position (σ) can be roughly estimated by multiplying DOP with a pseudorange errors (σ_{UERE}) as Equation 2.3.

$$\sigma = \mathrm{DOP} \cdot \sigma_{UERE} \tag{2.3}$$

2.2 PERFORMANCE OF STAND-ALONE GNSS

The common parameters for indicating GNSS performances are accuracy, availability, integrity, and continuity (Hofmann-Wellenhof et al., 2008, Kaplan and Hegarty, 2006). These parameters are commonly used by GNSS community to measure the overall performance of GNSS(s).

Accuracy is defined as the nearness of an estimated position to the actual or true position. It is usually presented as a standard deviation of errors together with a confidence level of indicating the probability of the given value. For example, SPS of GPS has a horizontal accuracy ≤ 9 m (95%).

Availability is the percentage of time that GNSS services are usable taking into account all the outages for every time step within a specified coverage area. The service is considered available if certain performance criteria, such as accuracy and integrity, meet a threshold requirement. However, the definition and criteria can be varied depending on the application's requirements. For example, Feng (2003) defined the availability of GNSS solutions by setting

the accuracy threshold at 20 m (95%) of horizontal errors and reported that GPS is available 90 % of the time when operating in open sky areas.

Continuity is the probability that GNSS service can maintain its performances without non-schedule interruptions during an intended operation, presuming that it is available at the beginning of the operation. Different applications may require different levels of GNSS continuity. For example, land navigation systems/services require lower level of continuity than aircraft navigation systems.

Integrity relates to the level of trust on the correctness of the information provided by the GNSS service. It is the ability that a navigation system/service can give timely and valid messages (known as an alarm) to users when the system/service should not be used for an intended period of time. Integrity is usually presented as the time difference between the beginning of service failure and the time the warning message is sent to the user. The integrity information is crucial for safety-of-life applications.

2.3 REQUIRED NAVIGATION PERFORMANCE

The RNP term was originally defined by International Civil Aviation Organization (ICAO) in 1983 as "a statement of the navigation performance necessary for operation within a defined airspace". Since then, the concept of RNP has been extended to cover other modes of transportation, such as marine (Rainbow and Clarke, 2002) and vehicle (Ochieng et al., 1999, Quddus, 2006), and also LBSs (Filjar et al., 2004). Typically, RNP is the quantitative criteria, at the application level, for specifying required levels of navigation performance rather than measuring the performance of particular geo-positioning sensors. The parameters used to define RNP criteria (i.e., availability, accuracy, integrity, and continuity) are the same set as the parameters of GNSS performance. This is mainly because the performance of a navigation application is highly dependent on the performance of its positioning module.

To ensure that an application can deliver the services at a satisfactory level, the value of each RNP is defined as the application's requirements. Table 2.4 provides examples of different RNP levels for Advanced Transport Telematics (ATT), or Intelligent Transportation System (ITS), applications and services (Quddus, 2006).

Table 2.4. RNP for some ATT applications and services (Quddus, 2006)

Applications and Services	Accuracy (m, 2σ)	Availability (%)	Integrity (sec)
Navigation and route guidance	5-20	99.7	1-15
Automatic vehicle location (generic)	10-30	99.7	1-15
Emergency location	5-10	99.7	1-5
Public safety	10	99.7	1-15
Accident management	30	99.7	1-15
Collision avoidance	1	99.7	1-15
Rail position location	10-30	99.9	1-5
Automatic announcement of bus stops	5-10	99.7	1-15
Transit vehicle control and command	30-50	99.7	1-15

2.4 GNSS PERFORMANCE PREDICTION

The idea of predicting GNSS QoS was initiated in the GPSLoc framework by Karimi et al. (2004). This study has framed the QoS concept and its parameters required for GPS-based applications. The authors defined four parameters: visibility, accuracy, reliability, and flexibility, that are applicable to any application and user. The methodologies, models, and algorithms were discussed but only for measuring the visibility parameter.

Besides the GPSLoc framework that is closely related to this dissertation, there are other studies that predict GNSS performances to fulfill other purposes. Generally, the purposes of predicting GNSS performances reported in the literature mainly are: (a) system analysis, (b) location and time planning, and (c) positioning quality improvement.

System analysis evaluates simulated GNSS performances for a better understanding of various designs, factors, and issues impacting the overall GNSS performance. For example, O'Keefe et al. (2002) reported the improvement of combining GPS and Galileo on accuracy and reliability in an extreme masking environment. Seynat et al. (2004) developed the GNSS Simulation Tool (GST) for researchers and developers to provide specific answers to a number of GNSS performance related questions.

Location and time planning provides predicted GNSS performances at a certain location and time for planning GNSS data collection tasks. For example, Germroth and Carstensen (2005) developed a tool in ArcGIS for predicting satellite visibility that allows users to plan for the best GPS quality. Lee et al. (2007) developed a web-based application that visualizes GNSS availability for a given location and time.

Positioning quality improvement aims to reduce possible errors in the position calculation process of the receiver by utilizing predicted GNSS performances. For example, Lee et al. (2008b) developed a new approach for mitigating multipath signals using a GNSS prediction tools. Marais et al. (2005) develped a software tool to evaluate the availabilitly of satellites using images of visible sky taken by a camera, which helps identify the satellites that should be used for calculating a position in real time.

Much of the effort on predicting GNSS performances has been focused on developing simulation tools. Currently, a variety of simulation tools exist to simulate GNSS constellations and surrounding environments with the use of various data sources and implementation techniques. Generally, these simulation tools can be categorized into two groups: *2D-GNSS-Sim* and *3D-GNSS-Sim*. The main difference between 2D-GNSS-Sims and 3D-GNSS-Sims is that the former do not consider geographical surface of the Earth while the latter do. Table 2.5 summarizes the strengths and weaknesses of the two simulation types.

Table 2.5. Comparison of two simulation types for predicting GNSS performances

Types of Simulations	Strengths	Weaknesses
2D-GNSS-Sim	1. Large coverage 2. Applicable to any location and defined scenarios	1. Non-realistic environment settings 2. Imprecise prediction
3D-GNSS-Sim	1. Precise prediction 2. Good at local analysis	1. Limited to particular data sources 2. Limited to a certain geographic extent 3. Varieties of 3D data formats and processing tools 4. Computationally intensive due to 3D analysis

In general, 2D-GNSS-Sims are used by GNSS researchers and developers to perform comprehensive analysis of GNSS performances at regional or global level. They provide flexibility to customize configurations of GNSSs at all segment levels, such as defining configurations of satellite constellations, specifying types of receivers, modifying navigation messages, and changing the level of UERE values. They also have capabilities to compute many GNSS performance parameters, such as accuracy, availability, continuity, integrity, and reliability. 2D-GNSS-Sims can predict GNSS performances at any desired location or scenario (e.g., trajectory or a grid area). Typically, this type of simulations does not include 3D data for representing objects on the Earth surface but may allow the user to define the masking environment manually at each testing location (e.g., defining masking angle at 60 degrees for urban areas). This makes 2D-GNSS-Sims generic and applicable to any geographic area. However, due to the lack of realistic representation of local environmental structures, only coarse prediction can be achieved by 2D-GNSS-Sims. Moreover, 2D simulations require users to have a comprehensive knowledge of GNSS in order to operate the tools, create testing scenarios, and correctly analyze the predicted outputs for future planning. Examples of 2D-GNSS-Sims are Galileo System Simulation Facility (GSSF) (European Space Agency, 2007) , GST (Seynat et al., 2004), and VISUAL (Verhagen, 2002).

3D-GNSS-Sims have been developed, implemented, and utilized by those studies that aim for realistic and fine GNSS performance prediction (Beesley, 2002, Karimi et al., 2004, Lee et al., 2008a, Li et al., 2006, Lohani and Kumar, 2007, Suh and Shibasaki, 2007, Taylor et al., 2007, Taylor et al., 2005). Incorporating real-world representation allows the simulation to precisely determine the impact of the local environment on satellite visibility and signal propagation paths. This implies that the predictive performances are as close as possible to the real GNSS performances at a specific location and time. Common parameters considered by most simulations are availability of satellites and positional accuracy, which are the basic parameters of concern by most applications.

However, there are limitations that prevent current 3D-GNSS-Sims from being widely used. One is the availability of accurate 3D data to represent real-world objects. 3D data is currently not available for all areas. Thus, 3D-GNSS-Sims are able to predict GNSS performances only within coverage of the 3D data. Moreover, there is no standard on what 3D data models should be used in the simulation. The decision is entirely up to the developers and

availability of data sources. As a result, a diverse set of algorithms have been developed for acquiring and processing particular 3D data models. For example, Karimi et al. (2004) developed a new 3D data model, called XTIN, that can represent 3D objects, as Digital Elevation Model (DEM), based on a TIN surface model. Some studies chose to store and represent 3D data by using proprietary data formats. For example, the simulation tool developed by Bradbury (2007) and EnviNav (Rémy and Moura, 2007) used the Virtual Reality Modeling Language (VRML) model in a proprietary format to represent 3D objects. In addition, ArcGIS software package has been used in many studies to facilitate data visualization and perform 3D analyses (Beesley, 2002, Germroth and Carstensen, 2005, Taylor et al., 2007, Zhang et al., 2008).

Using 3D data models also adds complexity to the simulation, which slows down the prediction process. Currently, there is a void in the literature reporting on suitability of 3D-GNSS-Sims for real-time prediction.

Another limitation of the existing 3D-GNSS-Sims is that the outputs are in difficult-to-use formats, which are not informative to real-time navigation applications. This problem arises partly because predicting GNSS performances is a spatio-temporal problem, i.e., the predictive values vary based on location and time. Some studies provide predicted results at each fixed point over a period of time or at a fixed time (Bradbury, 2007, Karimi et al., 2004, Taylor et al., 2007). Some studies provide results in a grid format at a particular snapshot (Germroth and Carstensen, 2005, Steed, 2004, Zhang et al., 2008). Some studies predict parameters for a given trajectory but present the results as a series of points (Lee et al., 2008a, Li et al., 2006, Suh and Shibasaki, 2007) or an average over an entire route (European Space Agency, 2007). These output formats are designed to accommodate the offline process for performance analysis but do not support real-time navigation applications.

Table 2.6 provides a summary of existing GNSS performance simulation tools. In summary, although the three 2D-GNSS-Sims provide capabilities for manipulating GNSS configurations and computing various GNSS performance parameters, they cannot provide precise prediction at local scale. On the other hand, there are at least twenty 3D-GNSS-Sims but due to their shortcomings, discussed previously (i.e., the use of particular 3D data sources, non-standard 3D data models and formats, and computation complexity), they are not widely used in different geographical areas and for navigation purposes. Of these limitations, complexity of 3D simulations is a challenge to iGNSS QoS prediction. Thus, this impedes other 3D simulations to

provide predictive solutions to navigation applications in real time, which is one of the objectives of this dissertation.

Table 2.6. Summary of existing GNSS performance simulation tools

Literature	Simulation Tools	Predicting Parameters	Environment Models	Data Sources	Surface Models
2D-GNSS- Sims					
European Space Agency (2007)	GALILEO System Simulation Facility (GSSF)	- Accuracy - Integrity - NSP (Navigation System Precision) - Continuity - Availability - Degradation - Granularity	- Estimated UERE	-	-
Seynat et al. (2004)	GNSS Simulation Toolkit (GST)	- Availability - Continuity - Accuracy (DOPs) - Position error	- Ionospheric effect - Tropospheric effect - Multipath - Elevation Masking - C/N ratio	-	-
Verhagen (2002)	VISUAL, a MATLAB interface	- Availability - Accuracy DOPs - Reliability - Success rates	-	-	-
3D-GNSS-Sims					
Beesley (2002)	A GIS model (ArcGIS and ArcObjects)	- Accuracy	-	- LiDAR mass points - Building footprints	Grid Raster model (Vegetation canopy and buildings)
Bradbury (2007)	UCL software toolkit	-Availability	- Reflection, - Diffraction model - Fresnel zone analysis	- CAD - VRML	Virtual Reality City Models
Rémy, B. and Moura (2007)	EnviNav	- Availability - Continuity - Accuracy - Position error	- Estimated UERE	-	VRML or DXF data format

Literature	Simulation Tools	Predicting Parameters	Environment Models	Data Sources	Surface Models
Germroth and Carstensen (2005)	Satellite Viewsheds, based on ArcGIS capabilities	- Visibility	-	- DEM, LiDAR, or other surface models - 1-foot bare-earth elevation contours, - Building roof outlines and rides - Spot elevations	TIN
Jeannot et al. (2005)	Ergospace software, a simulation software for signal propagation	- Availability - Accuracy - Multipath	- Estimated UERE	- 3D Scenes: VRML, DXF or BD	3D model
Karimi et al. (2004)	GPSLoc, a framework	-Visibility	-	- DEM (30 m resolution) - 3D objects	XTIN = TIN + 3D objects
Kleijer et al. (2008)	-	- Availability - Accuracy (DOPs)	-	- Urban canyon model	3D model - assume flat area
Lee et al. (2008b)	A testing simulation	-Availability - Accuracy	-	- 30 cm DEM - 1m IKONOS satellite imagery	3D obstruction objects (grid cells)
Lee et al. (2007)	A web-based simulation	- Availability - Accuracy - Multipath	- pseudorange multipath error (Ray tracing technique)	- 1 m DEM - 1 m DSM	3D Model (TIN)
Li et al. (2006)	-	- Multipath	- Multipath (reflected ray)	- 1 m LiDAR raster DSM - 2D building footprints	TIN
Lohani and Kumar (2007)	A model for predicting GDOP along with the probability of occurrence	- Accuracy	-	- LiDAR points	Bare Earth Model (BEM)
Marais et al. (2000)	PREDISSAT	- Availability	-	- Image from a mono-camera stereovision	-

Literature	Simulation Tools	Predicting Parameters	Environment Models	Data Sources	Surface Models
Marais et al. (2005)	PREDIctive Software for Satellite Availability in the field of Transport (PREDISSAT)	- Availability	- Diffraction - Reflection (Analyze at signal level)	- Image from a mono-camera stereovision	-
Marais et al. (2007)	PREDISSAT	- Availability	- Diffraction - Reflection (Analyze at signal level)	- Image - 3D VRML model of Rouen	3D model
Steed (2004)	Satview	- Availability	-	- Estimated heights of buildings	3D block model
Suh and Shibasaki (2007)	A simulation	- Availability - Accuracy - Multipath	- Signal propagation model - Multipath (reflected and diffracted ray)	- 30 cm DEM - 1 m IKONOS satellite imagery	3D Model derived from the DiaMap, Drawing Exchange Format (DXF)
Taylor et al. (2007)	A software tool, based on ArcGIS capabilities	- Availability	-	- 1m LiDAR - Digital photogrammetry	TIN
Verbree et al. (2004)	A software tool, based on ArcGIS capabilities	- Availability	-	- Airborne laser-altimetry (point cloud) - Parcel boundaries (building, street, and canal)	- 3D city model - TIN for visibility calculation
Vrhovski et al. (2004b)	Satellite visibility tool, a tool for RUC (Road User Charging) scheme	- Availability	-	- DEM	- TIN
Zhang et al. (2008)	-	- Availability	-	- Aerial photography - Cadaster dataset - Building heights - Feature line work	3D model built by using ArcGIS with different datasets

3.0 IGNSS QOS PREDICTION

3.1 INTRODUCTION

As discussed in Chapter 2, the overall performance of GNSS can be measured through accuracy, availability, continuity, and integrity. These parameters are generically defined as statistics over a period of time at a location or across an area. However, they may not reflect the actual level of iGNSS QoS that the user will obtain at a specified location and time, especially in problematic areas. On the other hand, RNP parameters are defined as a set of criteria for navigation applications to guarantee overall navigation performance but they do not indicate quality of positioning solutions at a specific location and time. Thus, there is a need for a set of parameters that can indicate quality of GNSS positioning solutions for real-time navigation applications.

Existing GNSS simulation tools including 2D-GNSS-Sims and 3D-GNSS-Sims do not provide prediction information for real-time navigation as discussed in the previous chapter. 2D-GNSS-Sims are capable of computing several GNSS performance parameters and are applicable to various geographic areas but only coarse prediction information is provided. On the other hand, most existing 3D-GNSS-Sims are designed to predict only certain parameters (e.g., availability and accuracy) necessary for general applications. In order to provide iGNSS QoS prediction which allows recommendations based on GNSS QoS to navigation applications, a new methodology for predicting iGNSS QoS was developed.

In this chapter, we first define a set of parameters for iGNSS QoS prediction that are suitable for real-time navigation applications. This is followed by an overview of the iGNSS QoS prediction methodology. The details of the developed algorithms in the methodology are given in Chapters 4 and 5. Then, we clarify differences real-time navigation and real-time prediction defined in this research and the prediction approaches possible for computing iGNSS

QoS prediction are described. Last, a summary highlighting different characteristics of iGNSS QoS prediction and other existing simulation tools is provided.

3.2 IGNSS QOS PARAMETERS

To measure GNSS performances for real-time navigation systems/services, six parameters are defined: *visibility*, *availability*, *accuracy*, *continuity*, *reliability*, and *flexibility*. Their definitions are given in Table 3.1. These parameters are different from the parameters commonly used by the GNSS community and the ITS community, described in Chapter 2, in that they are defined and modeled to facilitate navigation functions.

Table 3.1. Parameters for iGNSS QoS

Parameter	Definition
Visibility	A reference to those satellites, out of all in view, that have a LOS to the receiver, or whose signal is able to propagate to the receiver
Availability	The ability that the iGNSS receiver can provide a positioning solution to the user
Accuracy	The degree to which iGNSS position is close to true location of the receiver
Continuity	A measure of how iGNSS can continuously maintain its availability and accuracy above the minimum requirement
Reliability	A value that indicates the degree to which iGNSS QoS, in terms of availability and accuracy, can meet the application's requirements
Flexibility	The ability to suggest alternative solutions when no solution at a given location and time can be obtained

Figure 3.1 shows the relationship among the six iGNSS QoS parameters and environmental and infrastructural factors impacting iGNSS QoS. The iGNSS QoS parameters can be categorized into three levels: point, segment, and recommendation. The information to derive each parameter flows from point level (lowest-level) to recommendation level (highest-

level). These parameters are not independent; the parameters at the higher levels provide information to the ones at the lower levels.

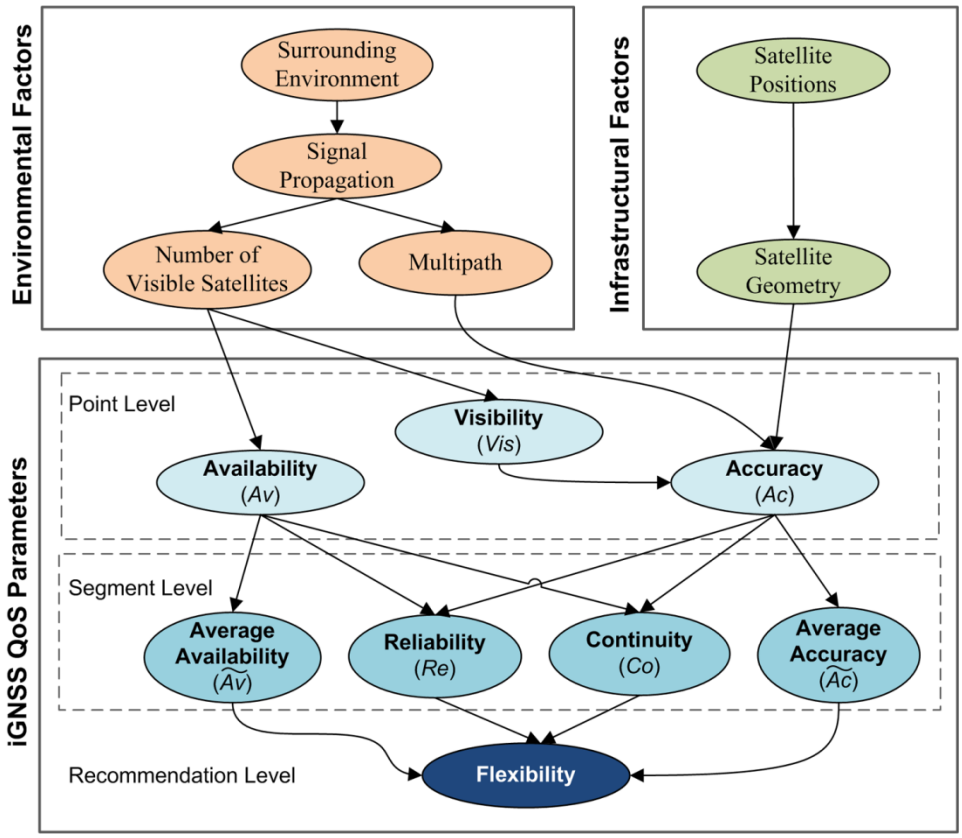


Figure 3.1. iGNSS QoS parameters and the relationships among them

The parameters at point level are visibility (*Vis*), availability (*Av*), and accuracy (*Ac*) predicted at a given location and time. The point-based parameters are the basis for computing the other parameters at segment (or polyline) level. The segment-based parameters are average availability ($\widetilde{Av}$), average accuracy ($\widetilde{Ac}$), continuity (*Co*), and reliability (*Re*) predicted for a given segment at a given time. The flexibility is the parameter at recommendation level that provides recommendation based on the point-based and segment-based parameters. Each of these iGNSS QoS parameters, except flexibility, can have a value between [0,1], where 1 indicates best quality and 0 worst quality.

The visibility parameter indicates the ability that the GNSS receiver can observe visible satellites out of all available satellites at a given location and time. Thus, *Vis* = 1 means that all available satellites are visible whereas *Vis* = 0 means none is visible. The main factor impeding the ability to observe all available satellites in view is the environment surrounding the receiver, which in turn impacts GNSS positional accuracy.

The availability parameter at point level indicates whether or not a GNSS solution will be available. This parameter is determined from visible satellites, see Section 4.3.1.2. If the number of visible satellites is more than four, the 3D positioning solution computed by a GNSS receiver is available (*Av* = 1), otherwise, it is unavailable (*Av* = 0).

The accuracy parameter at point level indicates the degree that the GNSS point is close to user's true location. It should be noted that iGNSS QoS accuracy is not the same as positional accuracy commonly measured by distance errors. Instead, the highest possible iGNSS QoS accuracy is 1 and the lowest is 0, see Section 4.3.1.3 for details. For navigation applications, we consider that any GNSS solution that deviates from user's true location more than 50 m has the lowest accuracy. The variation of the accuracy parameter is influenced by both environmental and infrastructural factors. The obstructed areas potentially occlude visibility of some satellite signals and possibly deteriorate the quality of some other received signals (i.e., attenuation and multipath effects). The geometrical arrangement of satellites is the infrastructural factor that can magnify positioning uncertainty as described in Chapter 2.

Availability and accuracy are the two parameters that are defined at both point and segment levels. At segment level, average availability ($\widetilde{Av}$) indicates how often GNSS solutions will be available on a given segment. Similarly, average accuracy ($\widetilde{Ac}$) indicates the average level of positional accuracy along the given segment. They are computed from a collection of point-based QoS predicted at points on the given segment, see Section 5.3 for details.

The continuity parameter indicates the maximum duration along a given segment that a GNSS receiver can continuously maintain available GNSS solutions that meet the minimum accuracy required by a navigation application. Its value can be computed through the prediction of point-based availability and accuracy, see Section 5.3 for details. *Co* = 1 means that the required GNSS solutions can always be obtained whereas *Co* = 0 means that the required GNSS solutions cannot be maintained along the course of the given segment due to unavailability or

low accuracy. A segment with *Co* = 0.5 indicates that about half of the segment with contiguous GNSS solutions will meet the requirement.

The reliability parameter indicates the probability that a receiver moving along a given segment can obtain availability over a minimum accuracy requirement. Reliability is different than continuity in the sense that reliability does not take into account continuity of GNSS solutions. Thus, it is possible that a segment may have high reliability but lower continuity due to occasional low GNSS QoS solutions. Similar to the continuity parameter, the reliability parameter can be determined from the prediction of point-based availability and accuracy, see Section 5.3 for details.

Unlike other parameters, flexibility does not provide QoS in terms of a value range [0,1] but it provides navigation applications with alternative navigation solutions or recommendations (e.g., alternative routes or times of travel) based on QoS prediction at segment level. Recommendations are provided whenever GNSS QoS of current planned routes cannot meet the application's requirements. Examples of route recommendations are given in Chapter 7.

3.3 METHODOLOGY

A methodology for predicting iGNSS QoS is developed to overcome the limitations of existing simulations while providing meaningful results to navigation applications and users. Figure 3.2 shows the proposed methodology which encompasses four modules: (i) a *segment sampling* module that samples points on route segments, (ii) a *point-based iGNSS QoS prediction* module that predicts QoS on sampled points, (iii) a *tracking-based iGNSS QoS prediction* module that summarizes point-based prediction on the route segments, and (iv) an *iGNSS QoS segmentation* module that divides each segment into similar QoS chunks. For short, we will refer to point-based iGNSS QoS prediction as point-based prediction and to tracking-based iGNSS QoS prediction as tracking-based prediction.

The methodology requires a route as an input in a form of a route, composing of a series of segments. Additional input includes starting time and QoS requirements. Although other forms of inputs are possible, such as a point or an area, this research focuses on prediction of QoS on route segments to support navigation applications. This is mainly because navigation

applications use networks, composing of nodes and lines, to represent real-world structures of traversable paths; for instance, road networks for car navigation and sidewalk networks for pedestrian navigation. A digital map database, containing a network and associate features (e.g., street name, street direction, and number of lanes), is a core component that supports most navigation functions, including map matching, routing, route direction, and tracking. Therefore, the expected output from this methodology is iGNSS QoS on each route segment representing the predicted value of each parameter. An overview of each of the four modules is described as follows.

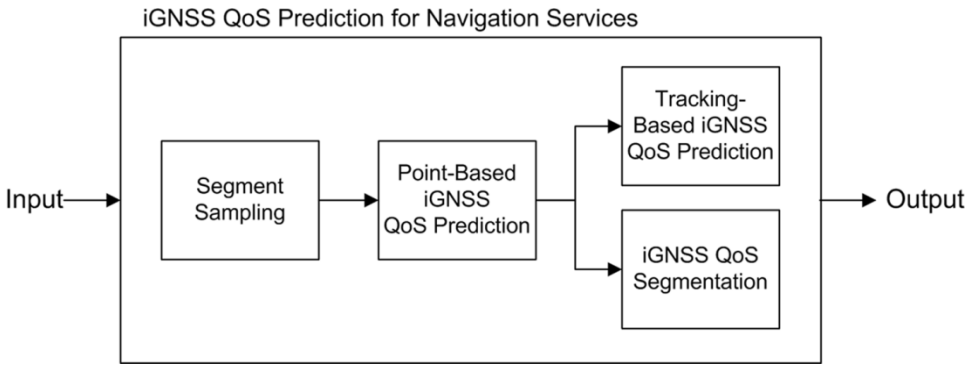


Figure 3.2. Methodology of iGNSS QoS prediction

Segment Sampling

The segment sampling module is responsible for selecting points on each segment of the given route. Since predicting iGNSS QoS is location- and time-dependent, the challenge is how to spatially and temporally sample points that best represent iGNSS QoS. One approach to this sampling is to consider every point on the route, but this is impractical due to a high demand of computing resources and the real-time constraint imposed by navigation applications. An alternative approach is to predict iGNSS QoS on a set of representative points along each segment. iGNSS QoS on representative points of a segment can be predicted by using the estimated time when the user passes each point.

In spatial analysis, the three main methods of spatial sampling in a geographic region are *simple random sampling*, *systematic sampling*, and *stratified sampling* (Rogerson, 2006). These methods can be applied to sample points on segments of a route.

Random sampling is the simplest method where points along the entire route are randomly selected, assuming each point has an equal probability of being selected. One main drawback of the random sampling method is that the sample points may not distribute evenly along the route, i.e., one segment may receive many points while others may receive few or no points. Unrepresentative sample points may significantly contribute to the inaccurate iGNSS QoS prediction for the entire route.

In the systematic sampling method with selection of an initial point on the route, other points with a certain interval are sampled. Two approaches are possible for this method. One is fixed-time interval between two points, e.g., sampling a point every second. Another is fixed-distance interval between two points, e.g., sampling a point every ten meters. The systematic sampling method, using either space or time interval, is most suitable in situations when there is no a priori knowledge of iGNSS QoS on the route.

The stratified sampling method is best suited in situations when there is a priori knowledge of iGNSS QoS on the given route. Information on iGNSS QoS could include statistics, such as means and variances, of iGNSS QoS prediction over a time period. The steps of the stratified sampling method are: (a) divide each segment into sub-segments according to QoS variation levels and (b) select points on each segment by using the systematic sampling method whose sampling interval is proportional to the variation level, i.e., sampling more points on sub-segments with high variation and fewer points on sub-segments with low variation.

In this dissertation, we chose to apply the systematic sampling method with a fixed-distance interval because (a) a priori knowledge of iGNSS QoS on different route types is not available and (b) a fixed-distance interval is absolute while in the fixed-time approach user's speed, which may vary on different segments, must be estimated in order to spatially sample points with a fixed-time interval.

Point-Based iGNSS QoS Prediction

The point-based prediction module predicts iGNSS QoS at each sample point. The details of this module are described Chapter 4. In summary, the module estimates three iGNSS QoS

parameters: visibility (*Vis*), availability (*Av*), and accuracy (*Ac*). The value of each of these parameters is within [0, 1] range where 1 is best quality.

The module comprises of a satellite orbit model, 3D data, and signal propagation models. Given a specific point and a time, this module simulates direct and non-direct LOSs between satellites and user's location. The three parameters are computed based on the simulated GNSS signals and the geometrical arrangement of the visible satellites.

3D data represents the surrounding environments, including man-made objects, vegetation, terrain, and other objects on the Earth's surface. In this dissertation, LiDAR point cloud data is used to build TINs for representing the Earth's surface. The high-resolution TINs constructed from LiDAR, while suitable for precise LOS calculation, contain vast amounts of data constraining real-time calculations. To reduce the amount of 3D data retrieval while ensuring correctness of LOS calculation, a strategy, called 3-zone LOS calculation, was developed.

A set of algorithms for determining signal paths is developed with the use of 3-zone LOS calculation. Three types of signal propagation are considered in this research: direct LOS, reflected LOS, and diffracted LOS.

Tracking-Based iGNSS QoS Prediction

Although fine granularity of iGNSS QoS prediction is possible through the point-based prediction approach, it is not well suited for navigation applications, especially for route calculations. This is because routing algorithms typically use a cost on each segment to compute optimal paths by minimizing the total cost between pairs of origin and destination addresses. To employ QoS prediction in routing algorithms, as a cost, a method for summarizing point-based QoS on segments is needed. Thus, the tracking-based prediction module employs point-based QoS prediction by calculating four iGNSS QoS parameters on each segment: average availability ($\widetilde{Av}$), average accuracy ($\widetilde{Ac}$), continuity (*Co*), and reliability (*Re*).

Details of computing each parameter are described in Chapter 5. With these parameters, at least four new routing criteria based on accuracy, availability, continuity, and reliability are possible for navigation applications, as discussed in Chapter 7.

iGNSS QoS Segmentation

While tracking-based iGNSS QoS prediction on segments provides information for route planning purposes, this information does not inform other navigation functions, such as map matching and real-time guidance, about the precise locations of poor iGNSS QoS on the segment. On the other hand, point-based iGNSS QoS prediction on the route can potentially overwhelm the application with redundant information. To address these problems, the iGNSS segmentation technique where it creates chunks (a chunk is a sequence of sub-segments with similar iGNSS QoS) is used.

iGNSS QoS segmentation has two main steps: (1) merging and (2) chunking. The merging step assigns the consecutive sampled points into sequences with a criterion that the variation of iGNSS QoS is within a threshold (τ). The chunking step constructs line geometry of chunks based on the points in each sequence and computes a statistical summary of iGNSS QoS for each chunk.

A new algorithm is developed for the merging step. The details of this module with the developed algorithm are described in Chapter 5.

3.4 REAL-TIME NAVIGATION VERSUS REAL-TIME PREDICTION

3.4.1 Real-Time Navigation

Real-time navigation is the ability that navigation systems/services can provide tracking information within a certain time constraint. Given that t_{app} is the navigation time constraint required to update the tracking information in an application and t_t is the tracking time required to determine user's new location, the following condition must be met:

$$\text{Real-Time Navigation: } t_t \leq t_{app}. \tag{3.1}$$

The navigation time constraint may vary depending on the requirement of navigation applications. For example, a car navigation application may require position updates every

second whereas a pedestrian navigation application may require position updates every five seconds.

The tracking time is the total execution time that a navigation system/service spends to identify user's current location, which can be formulated as follows:

$$t_t = t_p + t_{cs} + t_{mm} \tag{3.2}$$

where t_p is the time it takes to obtain new position data from positioning sensor(s),

t_{cs} is the time it takes to find a correct road/sidewalk segment from the map database,

$t_{mm}$is the time it takes to map match the obtained position data on the identified segment.

Typically, the longest waiting time for GNSS is the time it takes to acquire satellite signals, obtain navigation data, and calculate a position solution (called a fix), which is known as Time-To-First Fixed (TTFF). For GPS receivers, it can take up to 15 minutes for cold (or factory) start, and a few seconds for warm and hot start (Hofmann-Wellenhof et al., 2008). During the start of a trip or the transition between segments (at decision nodes), t_{cs} may take longer depending on a searching strategy used in the map matching module. Once the segment is correctly identified and the user travels on the segment, t_{cs} becomes zero.

If the condition in Equation 3.1 is not met, it is not possible for a navigation application to provide meaningful guidance in real time, in other words, navigation performance will be degraded due to the lack of updating new positions in time.

3.4.2 Real-Time iGNSS QoS Prediction

Real-time iGNSS QoS prediction is the ability that the predictive information can be provided to a navigation application within a certain time period before the user reaches the predicting location with the following condition:

$$\text{Real-Time iGNSS QoS Prediction: } t_c \leq t_{pred} \tag{3.3}$$

where t_{pred} is the available time period to provide iGNSS QoS prediction on a segment; in order words, it is the time it takes the user to travel from the current location to the beginning of the

predicting segment, and t_c is the time required to compute iGNSS QoS prediction for the predicting segment.

Time to predict, t_{pred}, QoS on segments farther away (in distance) from user's current location is longer than those near user's location.

The total computation time for computing iGNSS QoS prediction is as follows:

$$t_c = t_{sampling} + n \cdot \left(t_{pointQoS}\right) + t_{trackingQoS} + t_{chunks} \quad (3.4)$$

where $t_{sampling}$ is the time it takes to select sample points on a segment,

n is the number of sample points,

$t_{pointQoS}$ is the time it takes to calculate point-based iGNSS QoS on a sample point,

$t_{trackingQoS}$ is the time it takes to calculate tracking-based iGNSS QoS on a segment,

t_{chunks} is the time it takes to determine iGNSS QoS chunks on a segment.

Visibility calculation is the most time consuming component in the point-based iGNSS QoS prediction module. Longer segments require more time to compute visibility than shorter segments.

If the condition in Equation 3.3 is not met, the prediction information is useless for real-time navigation applications.

3.5 PREDICTION APPROACHES

To perform iGNSS QoS prediction, three approaches are possible:

Static prediction: providing iGNSS QoS for a given location (route or area) at a given time (fixed t_i = t for each p_i). Static prediction can be pre-computed days, weeks, months, or years in advance. However, like any other prediction (e.g., weather), the farther the prediction time, the more errors will be accumulated. The static prediction information is useful for locations under open sky or completely blocked environments where iGNSS QoS variations are minor at different times. The statistics of static prediction can also be used to decide whether or not dynamic prediction (defined below) should be performed.

Dynamic prediction: providing iGNSS QoS according to user's current and prospective locations. This approach must satisfy the condition of real-time prediction, where the prediction information must be available to the navigation application before the user reaches the desired location, otherwise, it is useless. This approach is more suitable in areas where iGNSS QoS fluctuates in short times.

Hybrid prediction: tradeoff between static and dynamic approaches. The information from the static approach can be used as initial knowledge to decide whether or not update is needed, and if update is needed, then the dynamic approach is performed. The tradeoff between the two approaches is dependent on the quality of prediction information required by applications, such as reliability. If highly reliable prediction is required, then the dynamic approach should be performed in those locations with high QoS variations.

3.6 IGNSS QOS PREDICTION AND EXISTING SIMULATION TOOLS COMPARISON

To highlight differences between iGNSS QoS prediction and existing simulation tools for predicting GNSS performance, described in Section 2.4, their characteristics are summarized in Table 3.2.

Table 3.2. Comparison between iGNSS QoS prediction and existing simulation tools

	iGNSS QoS Prediction	**Existing Simulation Tools**	
		2D-GNSS-Sim	**3D-GNSS-Sim**
Application	Navigation recommendations (e.g., route choices and routing)	GNSS performance analysis	• Location and time planning for stationary data collection • Positioning quality improvement analysis
Map Database	Road network Sidewalk network	N/A	N/A
Mode of Travel	Driving Walking	Not specific	Not specific
Prediction Units	Point Segments Chunks Trajectory	Point Mesh Trajectory	Point Trajectory
Parameters	iGNSS QoS: Visibility [0,1] Availability [0,1] Accuracy [0,1] Continuity [0,1] Reliability [0,1] Flexibility	GNSS performance: Availability Accuracy Integrity Continuity Availability Reliability	GNSS performance: Availability Accuracy Visibility Continuity
Surface/Object Representation	TINs+LiDAR	N/A	CAD DEM DSM TINs TINs+LiDAR Virtual reality city models XTIN
Real-Time Application Simulation	Yes	N/A	N/A
Computation Platforms	Standalone software possible to run on grids/clouds	Standalone software	Standalone software

4.0 POINT-BASED IGNSS QOS PREDICTION

4.1 INTRODUCTION

As a part of the iGNSS QoS prediction methodology described in Chapter 3, this chapter presents the details of the point-based iGNSS QoS prediction module for predicting positioning quality at a given location and time. The module composes of (a) an iGNSS simulation for determining signal paths of each individual satellite and (b) models for calculating the point-based iGNSS QoS parameters.

The iGNSS simulation includes a satellite motion model, 3D data modeled as TINs derived from LiDAR point cloud, and signal propagation models. A set of algorithms that use TINs are developed to determine a LOS of different signal paths, i.e., direct path, diffraction, and reflection.

To predict the iGNSS QoS parameters, models based on number of visible satellites are developed for calculating satellite visibility (*Vis*) and positional availability (*Av*) and a fuzzy-logic model is developed for estimating positional accuracy (*Ac*). The fuzzy-logic model takes into account geometrical arrangements of satellites (i.e., HDOP), multipath errors, and satellite visibility, with a set of rules to estimate the quality levels of positional accuracy.

An experiment was conducted and the predicted results were evaluated against reference data (GPS coordinates) at various locations. Evaluation results indicated that the proposed point-based prediction module can predict positioning quality with reasonably a high level of confidence in open sky locations and with some uncertainties in obstructed locations.

4.2 POINT-BASED IGNSS QOS SIMULATION

Factors impacting iGNSS QoS at a location can be categorized into two groups: systematic and dynamic. Systematic factors are assumed to be fixed which cannot be manipulated by users (e.g., atmospheric effects, receiver clock, satellite clock, and satellite positions). Dynamic factors include the surrounding environment (e.g., man-made objects, Earth's surface elevation, and vegetation) relative to user's location. Systematic factors can mostly be corrected by mathematical models for large-scale areas but dynamic factors affect positioning quality at local-scale areas. In this dissertation, user's surrounding environment is considered as the main dynamic factor in the point-based QoS prediction module. Figure 4.1 illustrates the architecture of this module which composes of an iGNSS simulation and the prediction models for the visibility, availability and accuracy parameter.

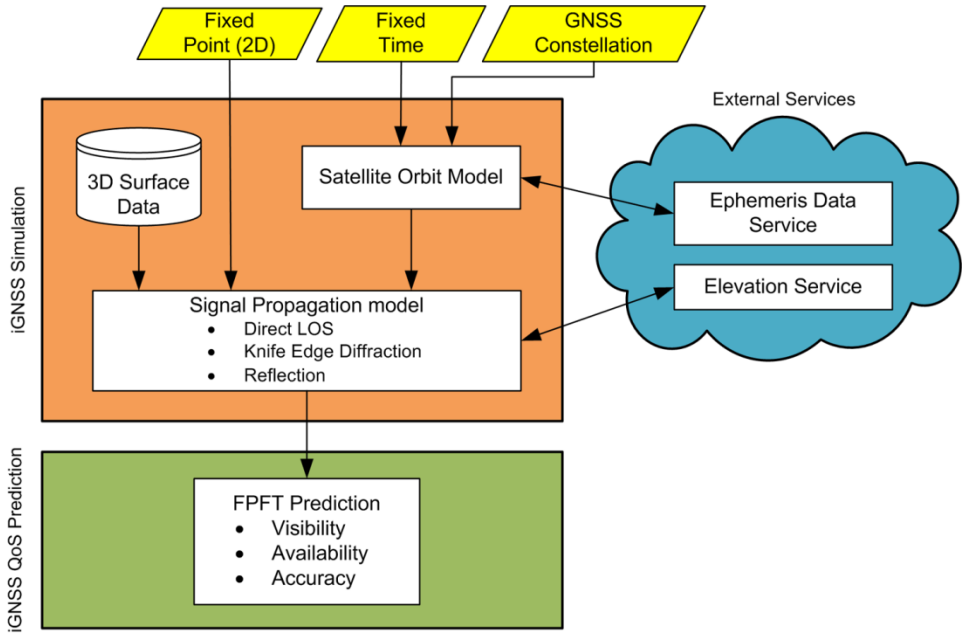


Figure 4.1. Architecture of the point-based iGNSS QoS prediction module

iGNSS simulation computes satellite positions and available signal paths at a given location and time. It has three components and data services described as follows.

4.2.1 Satellite Motion Model

Given a GNSS constellation, satellite positions can be computed at any given time using the Keplerian model (Parkinson and Spilker, 1996). This orbital model requires six basic parameters, reference time (T), longitude of the ascending node (Ω), eccentricity (e), inclination (i), argument of perigee (ω), and semi-major axis (a), to represent the Keplerian orbital elements.

To determine position and velocity of a satellite at any time instant, three types of data containing orbital information are available: almanac, broadcasted ephemerides, and precise ephemerides (Hofmann-Wellenhof et al., 2008). Almanac data provides adequate information to estimate satellite positions for acquisition and planning tasks. The estimated positions are correct within a few kilometers depending on the age of data. For instance, a two-week old GPS almanac usually causes a very small residual error in satellite position calculation (Driver, 2007). Broadcasted ephemerides are part of the satellite message broadcasted every half an hour and valid within a few hours or less. It allows the receiver to correctly estimate a satellite's position within 1 m. Precise ephemerides provided by the International GNSS Service (IGS) is the most accurate orbital information, about 0.05-0.2 m accuracy. However, the data are available with some latency, potentially introducing minimal error in position.

Our simulation currently uses ephemeris files for computing satellite positions. For GPS and GLONASS, the broadcasted data can be from GNSS data centers (e.g., IGS http://igscb.jpl.nasa.gov/). Galileo and Compass almanacs are currently not available. The design parameters of Galileo considered in Lee et al. (2008b) can be used to simulate its constellation.

4.2.2 3D Surface Data and Model

A 3D Geographic Information System (GIS) database represents real-world environments, including man-made objects, vegetation, terrain, and other objects on the Earth's surface. The quality of 3D data (i.e., accuracy, completeness, and resolution) greatly influences the correctness of satellite visibility calculations, which subsequently impacts the calculation of iGNSS QoS parameters. To adequately compute iGNSS QoS prediction, a 3D GIS database should contain complete and accurate data covering navigating areas and be in an open standard data format.

Given the aforementioned requirements, LiDAR point cloud is one potential source for constructing a high-resolution 3D surface model using TINs. LiDAR is a cost-effective technology that provides all significant Earth surface features such as buildings, trees, leaves and other man-made structures (Chen, 2007). LiDAR is expected to be widely available in the near future. Currently, several organizations such as CLICK (Stoker et al., 2006), OpenTopography[3], and PAMAP[4], have collected LiDAR point cloud data and made them available on the Internet. With the increasing number of providers, it is expected that highly accurate and large coverage LiDAR data will be periodically updated and maintained.

Taylor et al. (2007) have validated that TINs created from 1-m spacing LiDAR data and building footprints are appropriate for predicting satellites visibility, 90% correctness on predicting visible satellites. In our iGNSS simulation we also used 1-m spacing LiDAR data for creating TINs but with no building footprints; Oracle Database 11g was used to create and store TINs. For details of the LiDAR data source and TIN constructions used in this dissertation refer to APPENDIX A.1.

4.2.3 Data Services

Two data services are necessary for accurately predicting iGNSS QoS. One is a GNSS data service publishing the most updated ephemerides and almanacs of GNSS constellations. Another is an elevation data service for determining user's height at a given 2D point on the bare Earth's surface, such as the USGS's Elevation Query Web Service and the Google Elevation API, with the assumption that the user always travels on the ground.

4.2.4 Signal Propagation

The signal propagation module determines path of signals transmitted from each satellite. Three types of signal propagation are considered in iGNSS QoS prediction: direct, diffracted, and

[3] http://opentopo.sdsc.edu/gridsphere/gridsphere?cid=geonlidar
[4] http://www.dcnr.state.pa.us/topogeo/pamap/lidar.aspx

reflected LOS. From the signal path, satellite visibility and level of positioning errors due to multipath effect can be determined.

4.2.4.1 Direct LOS

Direct LOS is determined through the geometric relationship of a given observation point, a satellite position and surrounding objects. If no obstruction occurs along the LOS, the satellite has a direct LOS to the receiver. Considering the use of high-resolution TINs, we developed a strategy, called 3-zone LOS calculation, to reduce the amount of data retrieved while ensuring correctness of LOS calculations. In this strategy it is assumed that objects nearby the user have a higher chance of blocking the LOS than remote objects. Figure 4.2(a) illustrates the concept of the 3-zone LOS calculation, where direct LOS is first checked in Zone 1 then moving outward to Zone 3. The process terminates upon detection of a blockage.

The 3-zone LOS calculation requires three data layers with different granularity: *high-resolution*, *high-elevation*, and *bare-earth* TIN. A high-resolution TIN constructed from 1-m spacing LiDAR point cloud is used in Zone 1. A high-elevation TIN containing only objects with a certain height (h) above the ground (e.g., h = 20 m) is used in Zone 2. It should be noted that the value of h depends on the minimum distance of r_1 since this layer must include all possible objects that may occlude the lowest LOS at the cutoff elevation relative to user's location. For example, if the cutoff elevation (θ) is 10˚ and r_1 is 115 m, the minimum height of objects that could block LOS is 20.27 m. A TIN containing the bare Earth's surface is used in Zone 3. This is under the assumption that terrain has a high chance of occluding the LOS at remote distance from user's location. Figure 4.3 shows an example of LiDAR data used for constructing the three TIN data layers for the study area. For the steps of data preparation for each TIN data layer see APPENDIX A.2.

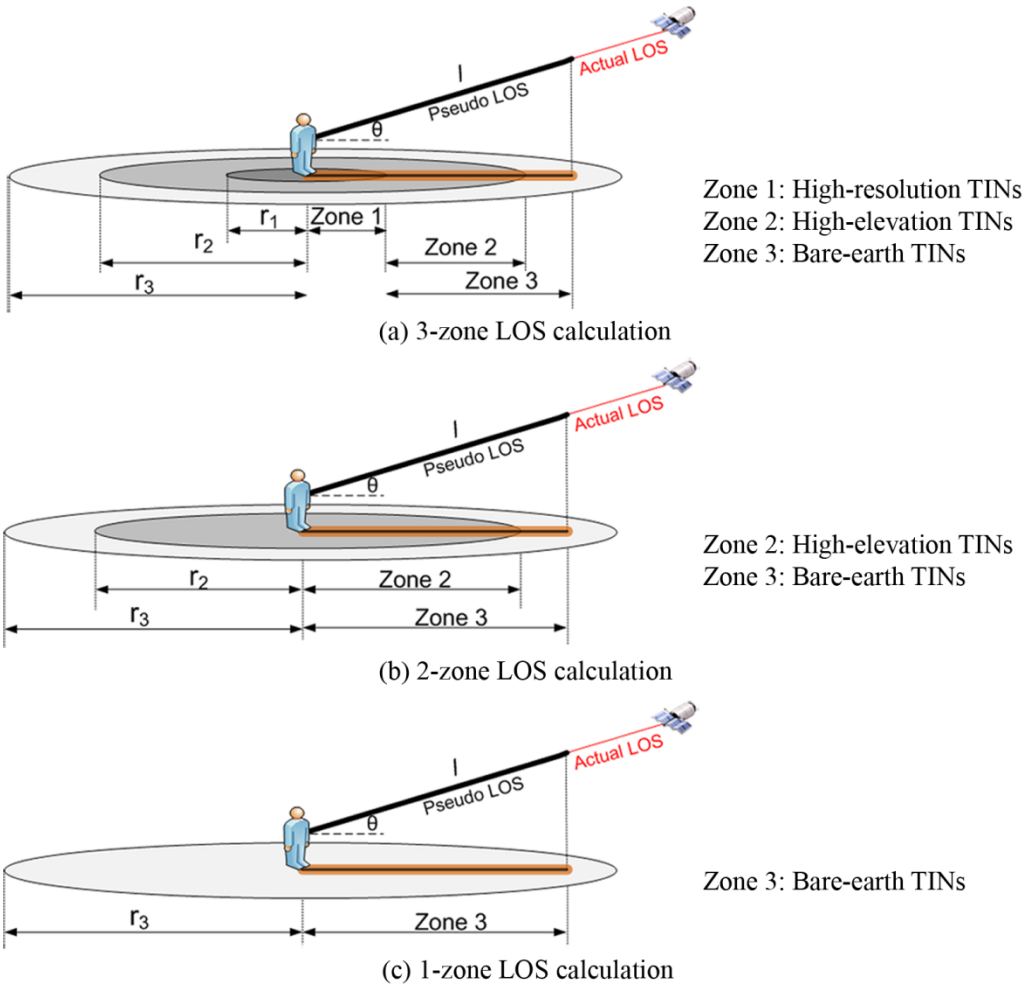
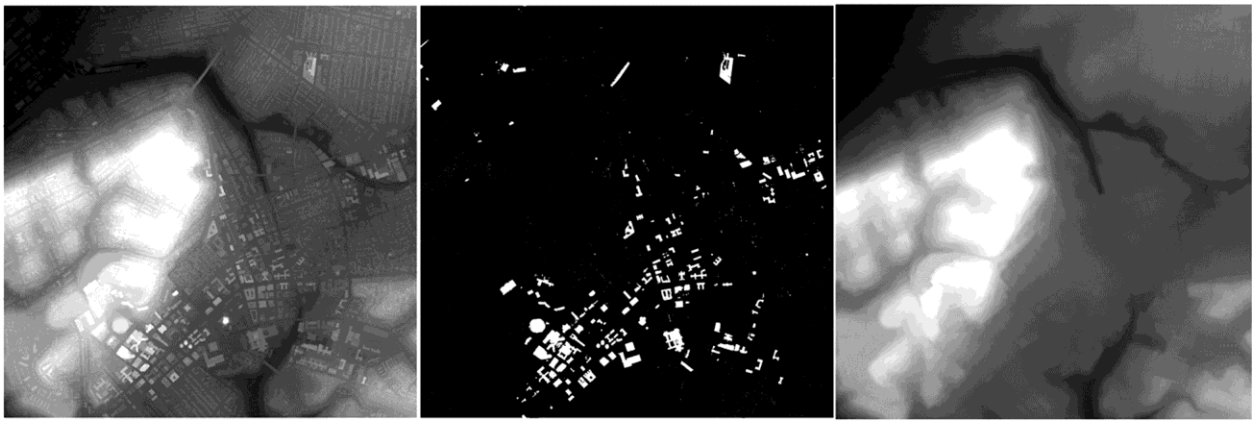


Figure 4.2. Different LOS calculations



(a) (b) (c)

Figure 4.3. LiDAR data for constructing the three TIN data layers used in the 3-zone LOS calculation: (a) high-resolution TIN for Zone 1 including all objects on the earth surface, (b) high-elevation TIN for Zone 2 including objects with 20 m above ground, and (c) bare-earth TIN for Zone 3

To further investigate the impact of granularities of TIN data layers on prediction results and computation performances, see Chapter 5, we modified 3-zone LOS calculation to 2-zone LOS calculation and 1-zone LOS calculation, as depicted in Figure 4.2(b) and (c). The 2-zone LOS calculation uses the high-elevation and bare-earth TINs while the 1-zone LOS calculation uses only the bare-earth TINs. Thus, the amount of 3D data details involved in prediction decreases as the strategy changes from the 3-zone to 1-zone LOS calculation.

Figure 4.4 describes the direct LOS algorithm between satellite's position (*sat_xyz*) and user's position (*user_xyz*) based on the 3-zone LOS calculation. Three TIN data layers are required for LOS in each zone where *TIN*[1] is high-resolution TINs for Zone 1; *TIN*[2] is high-elevation TINs for Zone 2; and *TIN*[3] is bare-earth TINs for Zone 3. The algorithm requires initial values for other configurations, including the length of the pseudo LOS (l), the horizontal width of the LOS buffer (w), and the distances to determine the three zones ($R = [0\ r_1\ r_2\ r_3]$) centered at user's location. If a direct LOS is found, the algorithm returns TRUE, otherwise FALSE. Examples of direct LOSs are shown Figure 4.8 and Figure 4.14 in yellow rays.

```
Input parameters:    user_xyz, sat_xyz
Output parameters:   los
directLOS (user_xyz, sat_xyz ) {
    los_3d ← Create a 3D LOS between user_xyz and sat_xyz with l distance from
user_xyz
    los_2d ← Project los_3d on horizontal plane
    2d_buffer ← Create a buffer area along los_2d with width w
    for each zone i = 1 to 3 do {
        r_in = R[i]      // inner radius
        r_out = R[i+1]  // outer radius
        if i = 3 then
            r_in = R[i-1]          // adjust inner radius for Zone 3
        ring_zone ← Create a ring buffer centered at user_xyz with radius r_out and r_in
        2d_buffer_zone ← Clipping 2d_buffer with  ring_zone
        T ← Clipping TIN[i] with 2d_buffer_zone          // results are a set of triangles
        for each triangle t_i in T do {
            if t_i intersect los_3d then
                    return los = FALSE         // no direct LOS found
        }
        i = i+1
    }
    // No intersection found
    return los = TRUE          // a directed LOS found
}
```

Figure 4.4. Direct LOS algorithm

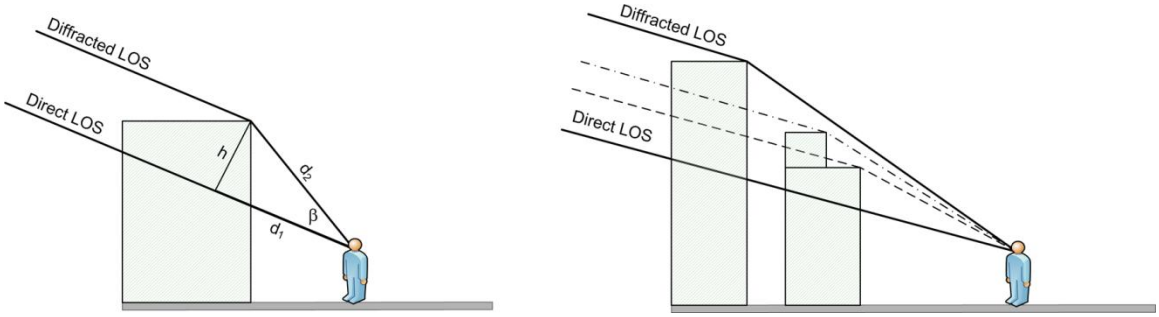
4.2.4.2 Knife-Edge Diffraction

Knife-edge diffraction occurs when a signal bends its incident angle around the edge of the object, as shown in Figure 4.5(a). As a result, the area behind the object is not completely shadowed but causes the diffracted signals to travel longer than the direct LOS. To determine whether the receiver can track a diffracted signal, the expected signal loss (J) is computed from the geometry between the signal and the object defined as a Fresnel diffraction parameter (Hannah, 2001):

$$\nu = h \cdot \sqrt{\frac{2}{\lambda \cdot d_2}} \tag{4.1}$$

$$J(\nu) = 6.4 + 20\log\left(\sqrt{\nu^2 + 1} + \nu\right) \tag{4.2}$$

where ν is Fresnel diffraction parameter, $h = d_2 \sin\beta$, as depicted in Figure 4.5(a), and λ is the wavelength of received signal (e.g., 19 cm for GPS L1 frequency). The delay path caused by the diffracted signal is $d_2 - d_1$. Figure 4.6 shows the plot of the knife-edge diffraction loss for GPS L1 signals. In the direct LOS, the expected loss, $J(\nu)$, is at 0 dB. As ν or as propagation angle of diffracted signal increases, the expected loss increases which results in weaker diffracted signal. The receiver may not be able to track the diffracted signal if the loss is too high.



(a) (b)

Figure 4.5. Knife-edge diffraction: (a) geometry and (b) determination of the diffracted point

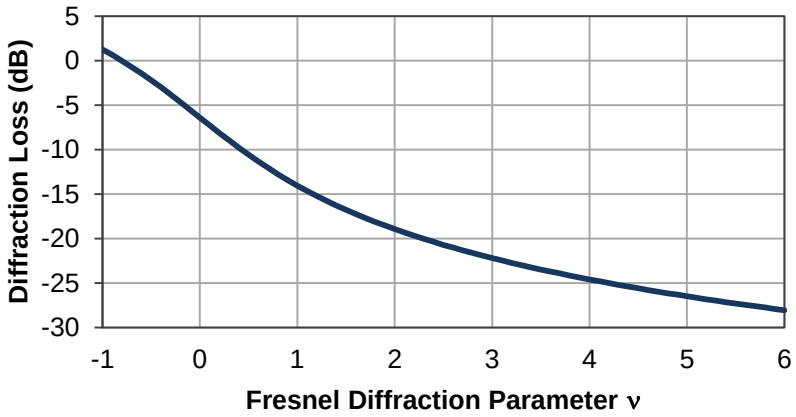


Figure 4.6. Knife-edge diffraction loss versus Fresnel diffraction parameters ν

Three assumptions are made in our iGNSS simulation: (a) the signals with loss ≤ 30 dB can be used, (b) only one-time diffraction is considered, and (c) adjacent triangles of TINs with differences of plane direction less than 60° are on the same surface. A diffracted LOS algorithm based on the use of high-resolution TIN was developed. This algorithm is invoked when no direct LOS is detected. Figure 4.5(b) illustrates the logic of the algorithm. It first finds the nearest obstacle whose top edge has a clear LOS to the observation point and to the satellite. It then checks to see if the expected loss of the signal is less than 30 dB.

Figure 4.7 shows the detailed algorithm for determining a diffracted LOS between satellite's position (*sat_xyz*) and user's position (*user_xyz*). Three TIN data layers are required for determining possible diffraction points and its LOS in each zone where *TIN*[1] is high-resolution TINs for Zone 1; *TIN*[2] is high-elevation TINs for Zone 2; and *TIN*[3] is bare-earth TINs for Zone 3. The algorithm requires initial values for other configurations, including the length of the pseudo LOS (l), the horizontal width of the LOS buffer (w), the distances to determine the three zones ($R = [0\ r_1\ r_2\ r_3]$) centered at user's location, and the maximum signal loss (max_J_v). If a diffracted LOS is found, the algorithm returns TRUE with the diffracted point (P_{dif}) and the distance delay ($Delay_{dif}$), otherwise FALSE. Figure 4.8 shows an example of diffracted LOSs in magenta rays.

Input parameters: *user_xyz, sat_xyz*
Output parameters: *los*, P_{dif}, $Delay_{dif}$
diffractedLOS (*user_xyz, sat_xyz*) {
    los_3d ← Create a 3D LOS between *user_xyz* and *sat_xyz* with *l* distance from *user_xyz*
    los_2d ← Project *los_3d* on horizontal plane
    2d_buffer ← Create a buffer area along *los_2d* with width *w*
    $h_{max} = -\infty$
    p_{dif} = *user_xyz*
    for each zone *i* = 1 to 3 **do** {
        $r_{in} = R[i]$ // inner radius
        $r_{out} = R[i+1]$ // outer radius
.   **if** *i* = 3 **then**
            $r_{in} = R[i-1]$ // adjust inner radius for Zone 3

. *ring_zone* ← Create a ring buffer centered at *user_xyz* with radius r_{out} and r_{in}

2d_buffer_zone ← Clipping *2d_buffer* with *ring_zone*

. *T* ← Clipping *TIN*[i] with *2d_buffer_zone*

dif_sat_los ← Create a 3D LOS between p_{dif} and *sat_xyz*

$t_{nearest}$ ← Find the nearest triangle in *T* that intersects with *dif_sat_los*

while ($t_{nearest}$ exists) {

h_{max} ← Compute the highest elevation of $t_{nearest}$ that its projection is on *los_2d*

T_{adj} ← Select adjacent triangles of $t_{nearest}$ in *T* where its plane direction is similar to $t_{nearest}$ (difference in direction plane < 60°)

p_{dif} ← the highest point on t_i

if T_{adj}.length > 1**then**

for each triangle t_j in T_{adj} **do** {

h_{max_adj} ← Compute the highest elevation of t_j that its projection is on *los_2d*

if $h_{max\_adj} > h_{max}$ **then**

$h_{max} = h_{max\_adj}$

p_{dif} = the highest point on t_j

}

dif_sat_los ← Create a 3D LOS between p_{dif} and *sat_xyz*

if any triangle in *T* intersects *dif_user_los* **then**

$t_{nearest}$ ← Find the nearest triangle in *T* that intersects with *dif_sat_los*

else

$t_{nearest}$ = NULL

}

J_v = Compute the expected loss // Equation 4.2

if $J_v < max\_J_v$ **then**

if *directLOS* (p_{dif}, *sat_xyz*) = TRUE and *directLOS*(*user_xyz*, p_{dif}) = TRUE **then**

return {*los* = TRUE,

$P_{dif} = p_{dif}$,

$Delay_{ref}$ = Compute the difference length between the actual 3D LOS (*user_xyz* to *sat_xyz*) and the diffracted LOS (*user_xyz* to p_{dif} to *sat_xyz*)}

else

return {*los* = FALSE,

P_{dif} = NULL,

$Delay_{ref}$ = NULL}

```
}
```

Figure 4.7. Diffracted LOS algorithm

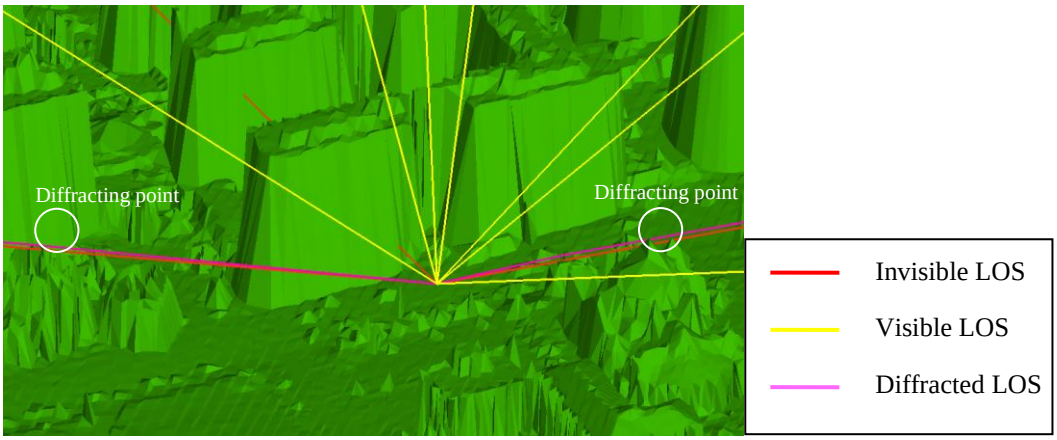
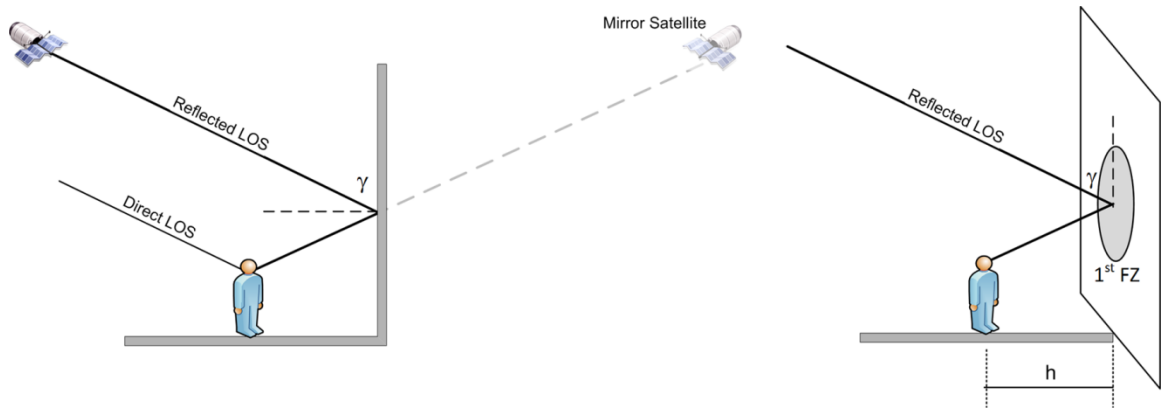


Figure 4.8. Example of diffracted LOSs

4.2.4.3 Reflection

Reflection occurs when the signal is reflected from a surface which makes it take alternative paths rather than direct LOS (Ray, 2000). The additional distance of pseudoranges adds up to positioning errors. In addition, the reflected signal can arrive in combination with a direct LOS signal, which causes multipath effect. The multipath effect has been comprehensively studied and modeled by the GNSS community from various perspectives, including geometry relationships, signal characteristics, and surface characteristics (Hannah, 2001, Ray, 2000). Our simulation considers only geometrical relationship among satellites, user, and surrounding objects that cause reflection, as depicted in Figure 4.9(a).



(a) (b)

Figure 4.9. Reflection (a) geometry and (b) Fresnel zone area

Fresnel Zone (FZ) analysis was applied to determine the regions of the reflecting surface (CCIR, 1986). It is generally accepted that the first FZ, containing only in-phase signals, contributes most to the reflected incident signal when the reflecting surface is much larger than the first FZ. The dimensions of the first FZ for reflection are a function of signal frequency, incidence angle, and perpendicular distance from surface to the receiver, as illustrated in Figure 4.9(b). The dimension of the first FZ can be computed as:

$$Radius:\ R_F = \sqrt{\lambda \cdot \frac{h}{\sin\gamma}} \tag{4.3}$$

$$Semi\text{-}major\ axis:\ L_{SMA} = \frac{R_F}{\sin\gamma} \tag{4.4}$$

where λ is the wavelength of received signal. In order to reflect the strong-strength signal, the reflecting surface should have an area larger than the first FZ.

To determine reflected signals using high-resolution TINs, the following assumptions are made: (a) only one-time reflected signals are considered, (b) all surfaces are capable of reflecting signals with the same reflection coefficient, and (c) adjacent triangles with differences of plane direction < 60° are considered as the same surface.

A reflected LOS algorithm based on the use of high-resolution TIN was developed, see Figure 4.10. Only the high-resolution TINs are required in this algorithm; *TIN*[1] is high-

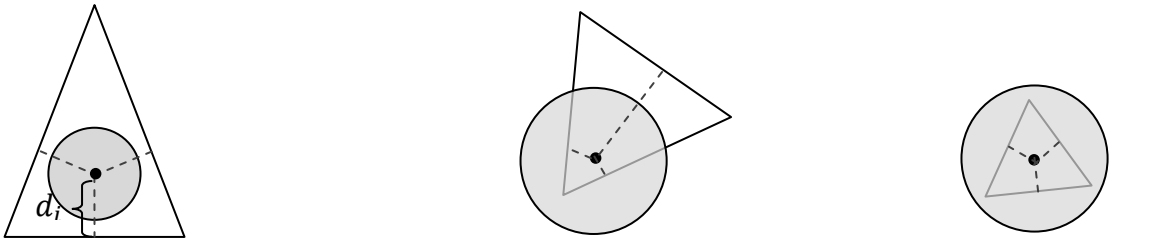
resolution TINs. The algorithm also requires an initial value of the radius distance surrounding user's location (r_{ref}) to limit the search of reflected signal paths within a reflection zone. If a reflected LOS(s) is found, the algorithm returns TRUE with the reflected point(s) (P_{ref}) and the distance delay(s) ($Delay_{ref}$), otherwise FALSE.

Input parameters: *user_xyz*, *sat_xyz*
Output parameters: *los*, P_{ref}, $Delay_{ref}$
reflectedLOS (*user_xyz*, *sat_xyz*) {
    2d_buffer_zone ← Create a horizontal buffer centered at *user_xyz* with radius r_{ref}
    T ← Clipping *TIN*[1] with *2d_buffer_zone*
    //Only vertical triangles are potentially blocked the LOS
    T_v ← Filter out triangles with plane direction < 30°
    for each triangle t_i in T_v **do**{
        p_{ref} ← Compute a point on t_i that reflects the LOS from *user_xyz* to *sat_xyz*
        if p_{ref} exists **then**
            L_{sma} ← Compute semi-major axis of the 1st Fresnel zone //Equation 4.3-4.4
            if *FZdetermination* (p_{ref}, L_{sma}, T_v, t_i) = TRUE **then**
                if *directLOS*(*user_xyz*, p_{ref}) = TRUE and *directLOS*(p_{ref}, *sat_xyz*) = TRUE **then**
                    $P_{ref}[j] = p_{ref}$
                    $Delay_{ref}[j]$ = Compute the difference length between the actual 3D LOS (*user_xyz* to *sat_xyz*) and the reflected LOS (*user_xyz* to p_{ref} to *sat_xyz*)
    }
    if P_{ref}.length > 0 **then**
        return {*los* = TRUE, P_{ref}, $Delay_{ref}$}
    else
        return {*los* = FALSE}
}

Figure 4.10. Reflected LOS algorithm

Because high-resolution TINs compose of a large amount of small triangles, a technique is needed to logically construct a composing surface from these small triangles in order to

determine the coverage of the first FZ area. There are three possible cases that the FZ will lay on the reflecting triangle, as shown in Figure 4.11. In Case 1, the triangle covers the largest size of the FZ, which can be determined by the nearest distance from the reflected point to all edges of the triangle (d_i), $d_i > L_{sma}$. Thus, the triangle has enough space to reflect the signal. Case 2 and Case 3 require composing triangles to cover the FZ.



Case1: FZ within a triangle

All $d_i > L_{sma}$

Case 2: FZ partly within a triangle

At least one $d_i > L_{sma}$

Case 3: FZ cover a triangle

All $d_i < L_{sma}$

Figure 4.11. Cases of FZ coverage on high-resolution TINs

An algorithm for determining the reflecting area was developed. The idea in this algorithm is to find a composing surface that covers the FZ, extract the outer edges of the composing surface, and then check if all d_i are larger than L_{sma}. Figure 4.12 shows the algorithm in detail. It takes four input parameters: the triangle containing the reflected point (t_{ref}), the reflected point on a triangle (p_{ref}), the semi-major axis of the 1st FZ (L_{sma}), and a set of the triangles on the same direction plane (T_v). If the composing surface covers the 1st FZ, meaning that this surface is capable of reflecting strong signals, the algorithm returns TRUE, otherwise FALSE.

Input parameters: p_{ref}, L_{sma}, T_v, t_{ref}
Output parameters: *reflectable*
FZdetermination (p_{ref}, L_{sma}, T_v, t_{ref}) {
D_1 ← Compute the nearest distance to all edges of t_{ref}

if all d_i in D_1 are greater than L_{sma} **then** // Case 1

return *reflectable* = TRUE

else // Case 2 and Case 3

FZ_area ← Create a buffer centered at p_{ref} with radius L_{sma} on the t_{ref} plane

// Find triangles on the same surface around the reflected point with the 1st FZ

for each t_i in T_v **do {**

if t_i intersects with *FZ_area* and t_i has plane direction similar to t_{ref} (difference of plane direction < 60°) **then**

E_{all}[k] ← all edges of t_i

}

// Find the outer boundary of the composing surface

f**or** each e_k in E_{all} **do** {

if e_k appears only once in E_{all}

$D[m]$ ← Compute the distance to e_k

}

if all d_m in D are greater than L_{sma} **then**

return *reflectable* = TRUE

else

return *reflectable* = FALSE

}

Figure 4.12. Reflecting surface determination algorithm

An example of determining the reflecting surface in Case 2 is given in Figure 4.13. The outer edges are e2, e4, e6, e7, e8, and e9. The nearest distance to e7 and e8 are less than L_{sma}, thus the area of this composing surface is not sufficient to reflect the signal. Figure 4.14 shows an example of reflected LOSs in cyan rays calculated by the two algorithms described above.

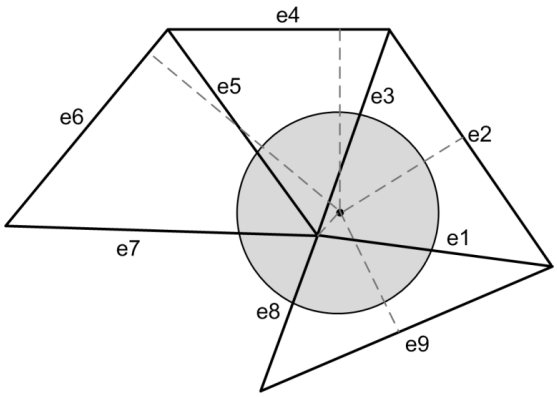


Figure 4.13. Example scenario of reflecting surface determination algorithm

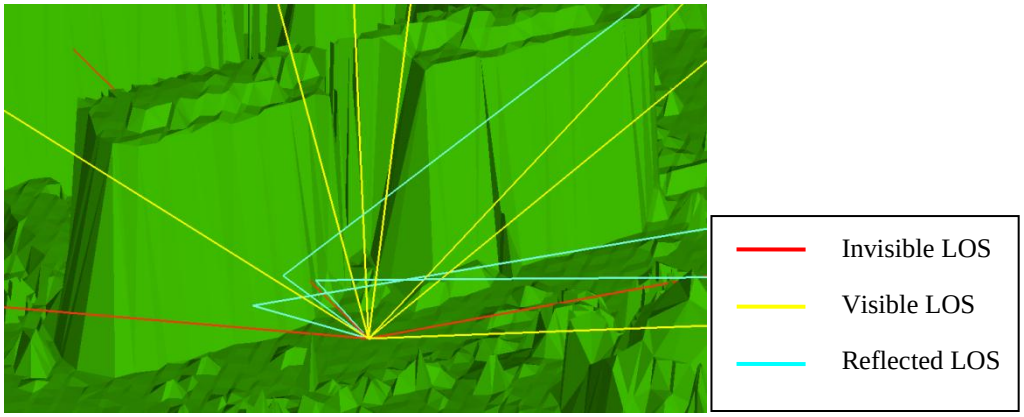


Figure 4.14. Example of reflected LOSs

4.3 POINT-BASED IGNSS QOS PARAMETERS

After obtaining satellite positions and signal paths at a given position and time, the three iGNSS QoS parameters (visibility, availability, and accuracy) can be computed as follows.

4.3.1.1 Visibility

In this research, we define visibility (*Vis*) as a quantitative value that indicates the ability to acquire satellites, out of all satellites in view, at a given location and time. Note that *Vis*

should not be confused with visibility of individual satellites. *Vis* captures the impact of the surrounding environment which in turn influences accuracy level. It is expressed as:

$$Vis = \frac{w_1 \cdot (N_{direct}) + w_2 \cdot (N_{diffracted}) + w_3 \cdot (N_{reflected})}{N_{direct} + N_{diffracted} + N_{reflected} + N_{blocked}} \quad (4.5)$$

where N_{direct}, $N_{diffracted}$, $N_{reflected}$, and $N_{blocked}$ are the number of satellites with direct, diffracted, reflected, and blocked LOS, respectively; $w_1, w_2,$ and w_3 are weights for direct, diffracted, and reflected satellites, respectively, and $w_1 > w_2 > w_3$. In the case that more than one signal paths from the same satellite received by the receiver, only the path with best signal quality will be included in *Vis* calculation. We considered the signal quality in this order: direct LOS > diffracted LOS > reflected LOS. Thus, the satellite sets for different LOS types are mutually exclusive. The value of *Vis* is [0,1] where 1 indicates all satellites in view available and 0 indicates none of satellites available to the receiver at a given location and time. Generally, the value close to 1 is expected in open sky areas. In the current version of the simulation, $w_1 = 1$, $w_2 = 2/3$, and $w_3 = 1/3$, to systematically prioritize the weights for the three signal paths with equal weight interval, are used. However, these weights could be determined empirically to minimize the difference of the visibility parameter between actual GNSS and GNSS prediction.

4.3.1.2 Availability

The availability parameter indicates the availability of positioning solutions at a given location and time based on the number of visible satellites (N_{vis}), $N_{vis} = N_{direct} + N_{diffracted} + N_{reflected}$. Theoretically, 3D positions can be computed when at least four satellites can be observed. Equation 4.6 provides a criterion for the value of availability.

$$Av = \begin{cases} 1; & N_{vis} \geq 4 \\ 0; & otherwise \end{cases} \quad (4.6)$$

4.3.1.3 Accuracy

The accuracy parameter indicates the degree to which the iGNSS positioning solution is close to the true location of the receiver at a given time. Accuracy of the estimated positions depends on two factors: geometry of satellites and UERE. Geometry of satellites is measured by DOPs. HDOP, for 2D coordinates, is often used in land-based navigation applications. HDOP

can be computed by simulating geometrical arrangement of satellites relative to an observation point. UERE is a root-sum-square of several error sources including ionosphere, troposphere, measurement noise, ephemeris error, clock bias, and multipath delay (Hofmann-Wellenhof et al., 2008). Most of these errors are systematic and assumed to be fixed for a period of time while multipath dynamically changes with surrounding environments as user's location changes in navigation applications.

To measure multipath errors, we compute an estimated position error from the residue matrix of position estimation ($\Delta\mathbf{x}$) as follows:

$$\Delta\mathbf{x} = [x_E \quad x_N \quad x_U \quad x_t]^T = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}\Delta\boldsymbol{\rho} \tag{4.7}$$

where $\mathbf{A}$ is the design matrix and $\Delta\boldsymbol{\rho}$ is the pseudorange error matrix (Hofmann-Wellenhof et al., 2008). $\Delta\mathbf{x}$ can be $\mathbf{0}$ if no multipath exists. The horizontal residue is computed by $\sqrt{x_E + x_N}$ and is used for indicating the level of multipath errors. The pseudorange error (ρ_i) of each satellite due to multipath signals is computed by using a theoretical Delay Lock Loop (DLL) correlation function which is a function of Multipath-to-Direct Ratio (MDR), time delay, and phase of the multipath signal (Van Dierendonck et al., 1992). MDR of all multipath signals is set to 0.5 and the phase is fixed to 0° for the maximum constructive effect on the composite signal. The standard early-late DLL with a narrow chip space (0.1 chip space) was used. Thus, the pseudorange error of the composite signal is within 30 m for the case of GPS. Large pseudorange errors are also possible if a reflected signal is tracked rather than the direct signal.

Although the simulation is capable of predicting several signal types, there are other hidden factors (e.g., resolution and correctness of the surface model, materials of reflecting surface) that cause uncertainties on the prediction result. In our accuracy model, we use the visibility parameter to infer the impact of hidden environment factors with an assumption that low visibility is likely to cause low positional accuracy (this will be discussed in Section 4.4.4.5).

To combine factors impacting accuracy (i.e., HDOP, multipath errors, and visibility), a fuzzy logic model was developed. Fuzzy logic is flexible in computing, tolerant to imprecise data and contradicting information, and easy of handling non-linear function of arbitrary complexity. Thus, the fuzzy logic model was chosen to handle non-crisp quality levels of the factors impacting the accuracy. In addition, rule-based decision deems appropriate for estimating the accuracy prediction.

The fuzzy logic model was designed based on a three-step process: (1) fuzzification of the input and output; (2) construction of a rule base; and (3) defuzzification of the output (Jang et al., 1997). This process is often referred to as a Fuzzy Inference System (FIS). In this work, a Sugeno-type FIS was chosen because its result is a quantitative number which can be used to indicate fine-scale levels of accuracy. Details of the developed model are as follows.

Step:1 Fuzzification of input and output

The input variables are HDOP, visibility (*Vis*), and multipath errors (*ME*) in meters whose values are obtained from the iGNSS simulation. The fuzzy subsets associated with each variable are: Good, Fair, and Poor for *HDOP* and *Vis*; High and Low for *ME*.

The Membership Function (MF) defines the degree to which an input belongs to a fuzzy set. The MFs of each input variable were initialized based on estimated values for different locations under various environment settings, i.e., open sky, moderately blocked, and completely blocked areas. For example, $1 \leq$ good $HDOP < 4$ is expected in open sky areas, $4 \leq$ fair $HDOP < 8$ is expected in moderately blocked areas, and poor $HDOP \geq 8$ is expected in blocked areas; $0.8 <$ good $Vis \leq 1$ is expected in open sky areas, $0.4 <$ fair $Vis \leq 0.8$ is expected in moderately blocked areas, and $0 \leq$ poor $Vis \leq 0.4$ is expected in blocked areas; and 0 m $\leq$ low $ME <$ 5 m is expected in moderately blocked areas, and 5 m $\leq$ high $ME \leq$ 30 m is expected in blocked areas.

The fuzzy output is the degree of obtaining a high accurate positioning solution at a particular location and time. A zero-order Sugeno fuzzy model is considered which takes eight constants for the output (denoted as *Ac*), *Ac* = 1.0, 0.9, 0.75, 0.6, 0.45, 0.3, 0.15, and 0.0 from excellent to poor positional accuracy, with equal interval except between 0.9 and 1.0. Each of the eight output constants is assigned to one of the fuzzy rules, explained as follows. Note that the predicted output does not directly indicate distance errors of points that will be obtained from the receiver but it gives an expected level of positional accuracy.

Step 2: Construction of rules for iGNSS QoS accuracy

The rules of the FIS for estimating iGNSS QoS accuracy are developed based on general knowledge about GNSS positional accuracy in terms of DOPs and observations on the relationships between predicted data (i.e., *Vis*, *HDOP*, and *ME*) and quality of GPS data at different testing sites under different environment settings. Examples of the relationships between statistical values of prediction data and that of GPS errors for the three input variables are shown in Figure 4.25. It is observed that a location with good HDOP and good visibility

obtains small positioning errors; this condition is defined in Rule 1 with the highest iGNSS QoS accuracy (*Ac* = 1). *ME* is not included in Rule 1 because at a location with good satellite visibility the multipath errors are minimal. Rules 1-7 are defined for different combinations of fuzzy levels for the three input variables where *Ac* is initially set at 0.15 intervals. Rule 8 indicates the impact of poor satellites geometry on GNSS positional accuracy. Regardless of values of visibility and multipath errors, high DOPs are the main sources of poor positional accuracy (*Ac* = 0).

The following rules are applied to the FIS:

1. If (*HDOP* is *Good*) AND (*Vis* is *Good*) then (*Ac* is *1.0*)
2. If (*HDOP* is *Good*) AND (*Vis* is *Fair*) AND (*ME* is *Low*) then (*Ac* is *0.9*)
3. If (*HDOP* is *Good*) AND (*Vis* is *Fair*) AND (*ME* is *High*) then (*Ac* is *0.75*)
4. If (*HDOP* is *Fair*) AND (*Vis* is *Fair*) AND (*ME* is *Low*) then (*Ac* is *0.6*)
5. If (*HDOP* is *Fair*) AND (*Vis* is *Fair*) AND (*ME* is *High*) then (*Ac* is *0.45*)
6. If (*HDOP* is *Fair*) AND (*Vis* is *Poor*) AND (*ME* is *Low*) then (*Ac* is *0.3*)
7. If (*HDOP* is *Fair*) AND (*Vis* is *Poor*) AND (*ME* is *High*) then (*Ac* is *0.15*)
8. If (*HDOP* is *Poor*) then (*Ac* is *0.0*)

Step 3: Defuzzification of the output.

The *min* method is used for the fuzzy AND operator to obtain a result for each rule, known as rule strength. Each rule has a unit weight. Weighted rule strength for each rule is referred to as the "degree of applicability, ω". The output of the FIS is the weighted average of the degree of applicability (ω_i) and the outputs (Ac_i). The final accuracy output is:

$$Ac = \frac{\sum_{i=1}^{8} \omega_i Ac_i}{\sum_{i=1}^{8} \omega_i}. \tag{4.8}$$

The initial FIS was tuned by Matlab's fuzzy logic toolbox which allows optimization of fuzzy MFs using an Adaptive Neural Fuzzy Interference System (ANFIS). The training data set is statistics of pairs of prediction data (i.e., *Vis*, *HDOP*, and *ME*) and actual GPS accuracy (Ac_{gps}) at different testing locations, shown in Table 4.3. The optimization method chosen to train the developed FIS was a hybrid between least squares and back-propagation. The trained FIS was tested by using pairs of prediction data and actual GPS accuracy at each timestamp (about 900 seconds for each testing location). In order to obtain the FIS with minimum prediction errors,

the initial MFs of input variables were empirically adjusted before training but the *Ac* constant of each rule was fixed. Figure 4.15 shows the trained MFs for the FIS used for estimating the accuracy parameter.

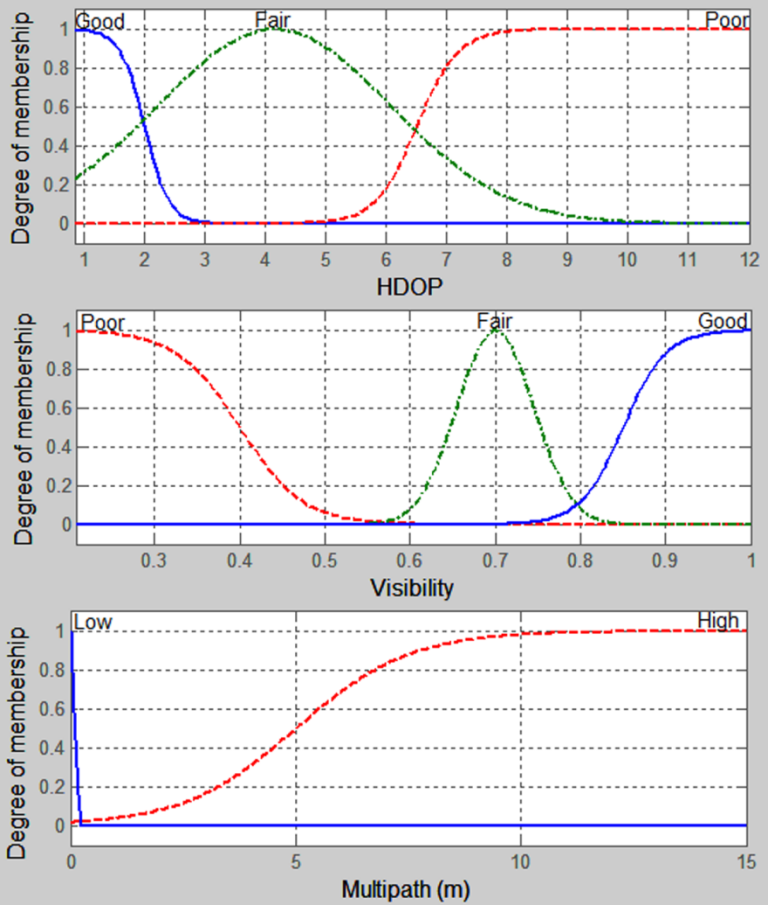


Figure 4.15. The membership functions for HDOP, visibility, and multipath error

4.4 EVALUATION

We evaluated (a) the correctness of satellite prediction by the iGNSS simulation including correctness of satellite prediction and blockage zone and (b) the robustness of the three iGNSS QoS parameters models including visibility, availability, and accuracy for indicating levels of iGNSS positioning quality in different surrounding environments. The predicted results were

compared with the actual GNSS data collected in the field. Since GPS is currently the only GNSS in full operation, only GPS QoS were evaluated.

4.4.1 Study Area and Reference Data

The study area covers the University of Pittsburgh's main campus, which has a mixture of environment settings. Three types of environment settings were considered in this evaluation: open sky, moderate or (partially blocked), and fully blocked. For each type, five locations were identified based on familiarity with the study area and the use of satellite imagery from Google Maps. In order to obtain similar quality level of the reference data for different environment settings, reference coordinates for the identified locations were digitized using Google satellite images © 2010 with a resolution better than 30 cm per pixel. This approach was validated based on another data set collected at 24 open sky locations in the study area by using post-processed DGPS. Each DGPS point was an average of five-minute data collected by Trimble's GeoExplorer®3 handheld GPS unit. The results showed that differences between DGPS points and digitized points were on average 1.591 m with standard deviation of 0.768 m; these translate into associated errors of about 1.6 m for the reference data. Figure 4.16 shows a map of the fifteen testing locations used for evaluation.

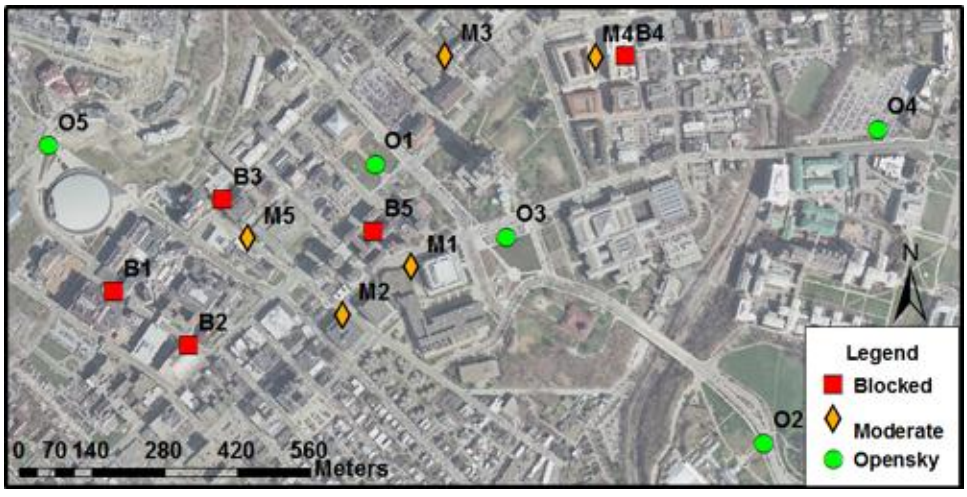


Figure 4.16. Testing locations

4.4.2 GPS Receivers and Data Collection

To emulate GPS receivers embedded in car navigation systems/services and navigation type receivers, where availability of positions is of particular importance, two high sensitivity GPS receivers with E-TEK EB85A and MTK II chipset, referred to as R1 and R2 to avoid a discussion of manufactures, were used. The reasons for using two different chipsets are to be able to (a) analyze the differences on positioning solutions calculated by different GPS receivers and (b) robustly tune the prediction models using different data sources. According to their specifications, R1 has -158 dBm sensitivity with 3.3 m Circular Error Probable (CEP) accuracy (no DGPS aided) and R2 has -165 dBm sensitivity with 2.5 m CEP accuracy (no DGPS aided). Both receivers, which provide output in National Marine Electronics Association (NMEA) format, were set to compute a position every second. At each location, the receivers were placed next to each other about 1.2 m above the ground and collected data for 15 minutes duration at two different times. Thus, a total of 30 data sets were collected.

4.4.3 Simulation Configurations

The simulation was run after the GPS data were collected, thus the exact time, or epoch, for each testing location was known. The elevation cutoff for satellites was set at 10°. The elevation of each location was obtained from the USGS's elevation web service. The three TIN data layers were generated from the LiDAR data obtained from the Pennsylvania State Data Center. The LiDAR data for Allegheny County was collected as part of the PAMAP[5] project in 2006 with a 1.4 m average point spacing (2 m maximum) and with a bare earth surface vertical accuracy of 18.5 cm RMSE. The signal propagation modules were set as follows. The cutoff distances in the 3-zone LOS calculation were set at r_1 = 115 m, r_2 = 1000 m, and r_3 = 3000 m and h = 20 m for the minimum height used in deriving the TIN data layer in Zone 2.

[5] http://www.dcnr.state.pa.us/topogeo/pamap/lidar.aspx

4.4.4 Results and Analyses

4.4.4.1 Correctness of Satellite Prediction

The evaluation matrix for this analysis is the confusion matrix or contingency table, traditionally used in assessment of classification modeling (Hastie et al., 2009). Figure 4.17 shows the evaluation matrix.

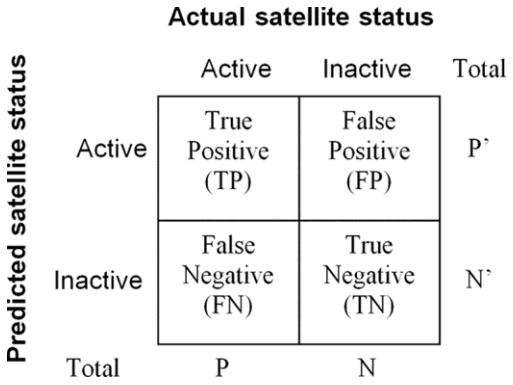


Figure 4.17. Evaluation matrix for satellite prediction

Active satellites are those having a LOS, otherwise they are considered inactive. Three statistical measures derived from the evaluation matrix are defined as:

$$\textit{satellite prediction accuracy}\ (\text{SPA}) = \frac{TP+TN}{P+N} \tag{4.9}$$

$$\textit{sensitivity or true positive rate}\ (\text{TPR}) = \frac{TP}{TP+FN} \tag{4.10}$$

$$\textit{specificity or true negative rate}\ (\text{TNR}) = \frac{TN}{TN+FP} \tag{4.11}$$

where P is the number of satellites predicted as active

N is the number of satellites predicted as inactive

TP (true positive) is the number of satellites predicted as active and actually active

TN (true negative) is the number of satellites predicted as inactive and actually inactive

FP (false positive) is the number of satellites predicted as active but actually inactive

FN (false negative) is the number of satellites predicted as inactive but actually active.

SPA indicates the accuracy of predicting active and inactive satellites. TPR is the rate of actual active satellites correctly predicted as active. TNR is the rate of actual inactive satellites correctly predicted as inactive. SPA, TPR, and TNR were calculated for each epoch. The average value of the three metrics was computed for each data set.

Figure 4.18 shows the statistics of the average SPAs and Figure 4.19 shows plots of the average of TPR versus that of TNP for different environment settings. The simulation predicted the active satellite accurately for open sky locations (SPA ≅ 96.5%) in both receivers. For GPS prediction, about 1.12 satellites were generally missed by the simulation due to the use of satellites at low elevation (< 10°) by the receivers. On average, 0.56 and 0.66 satellites at low elevation were used by R1 and R2, respectively. The simulation has a high sensitivity for predicting active satellites for open sky locations (TPR ≅ 88.6%), which mostly have direct and diffracted LOS.

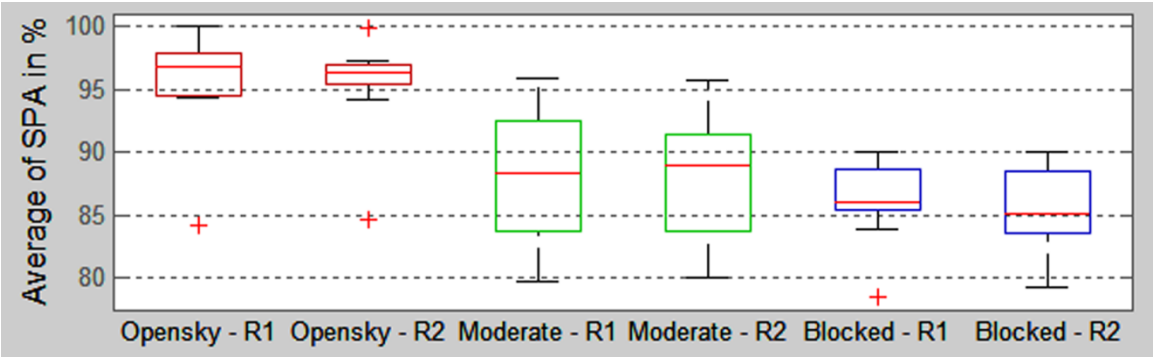


Figure 4.18. Average of Satellite Prediction Accuracy (SPA)

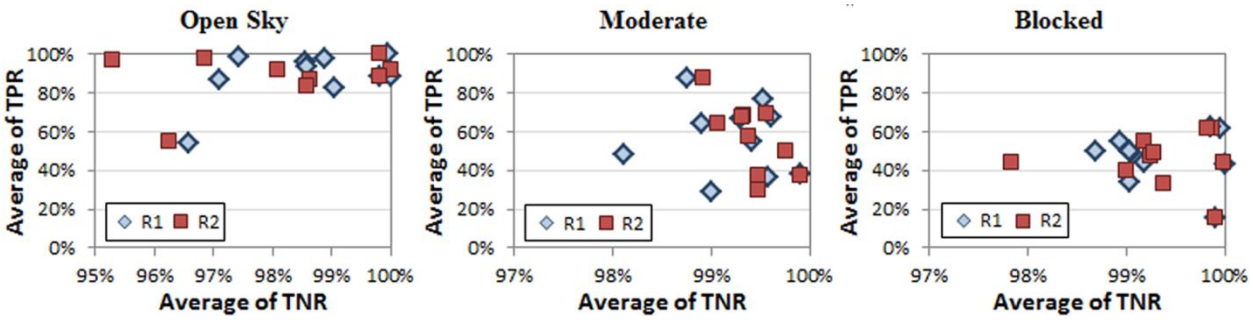


Figure 4.19. Sensitivity (TPR) and specificity (TNR) plots

For moderate and blocked locations, the simulation produced lower satellite prediction accuracies (SPA ≅ 88.7% and 85.6%). Large deviations for moderate locations can be observed due to a mixture of open and partially blocked sky view. The simulation also has low sensitivity on predicting active satellites (TPR ≅ 56.5% and 45.8% for moderate and blocked locations, respectively). In other words, it generally predicted a half of active satellites as inactive. There are at least two reasons for this. First, the signal propagation models could not compute all possible signal paths. Second, the high-sensitive receivers were able to track and use satellites with weak signal strengths, which increased the number of active satellites in problematic areas.

Note that reconfiguring the simulation to fit the actual active satellite is possible but not recommended because increasing the number of active satellites potentially decreases HDOP. Too optimistic predicted HDOP may not reflect the actual level of positional errors in problematic areas (this will be discussed in Section 4.4.4.5).

4.4.4.2 Blockage Zone

Since the purpose of 3-zone LOS calculation is to reduce the amount of TIN data used in visibility calculation, we analyzed effectiveness of using the zone strategy in LOS prediction. All predicted blocked LOSs of each epoch were recorded with (a) the zone that the visibility algorithm terminated and (b) the horizontal distance from the testing location to the intersect point of the LOS and an obstacle.

The results of all blocked LOSs found in each zone are shown in Figure 4.20 and the statistics of distances to intersected obstacles are shown in Figure 4.21. Under open sky condition, the number of blocked LOSs is less than that of moderate and blocked conditions. A

majority of blocked LOSs for open sky and moderate locations occurred in Zone 1, 83.6% and 98%, respectively, while all blocked LOSs for the blocked locations occurred in Zone 1. Within this study area, no LOS was blocked by terrain in Zone 3.

The blocked LOSs were intersected with an obstacle, on average, at 78.0, 36.1, and 26.3 m (horizontal distance) away from the receiver under open sky, moderate, and blocked conditions, respectively. The largest distance is 370.1 m occurred in Zone 2.

From this analysis, we concluded that Zone 1 is the most important area to determine blocked LOSs around the predicted location. The initial cutoff distance of Zone 1, r_1 = 115 m, was sufficient for determining LOSs under moderate and blocked conditions. Zone 2 is necessary for determining LOSs at farther distance for open sky and moderate conditions but the cutoff distance r_2 could be changed depending on the study area. Visibility calculation at Zone 3 may not be required if the study area is rather flat or small variation of terrain.

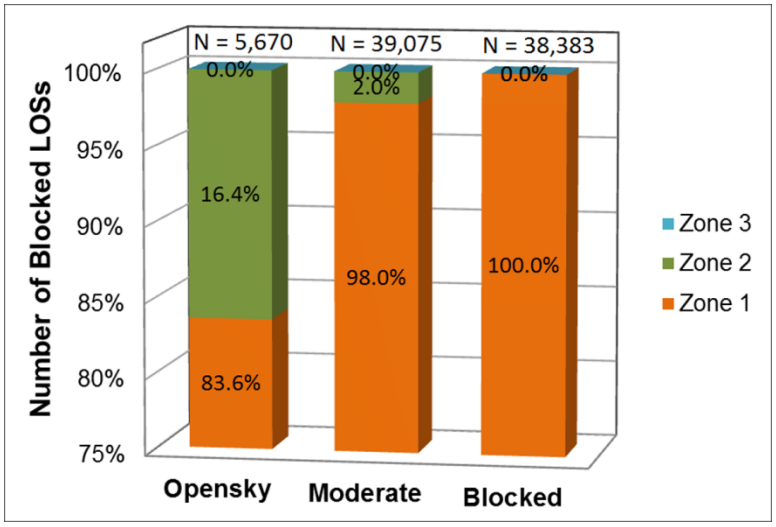


Figure 4.20. The number of blocked LOSs terminated within each of the three zones

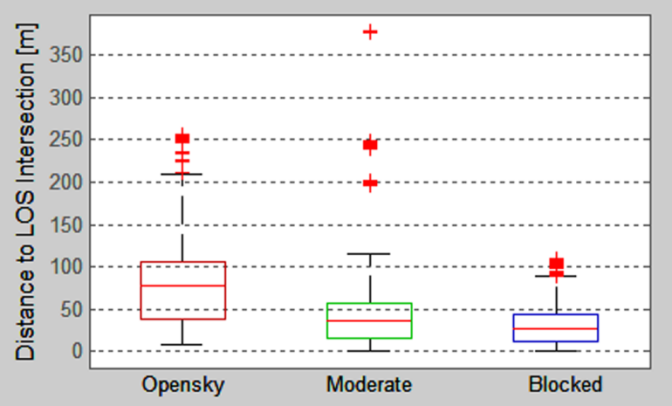


Figure 4.21. Statistics of distance to an obstacle for blocked LOSs

4.4.4.3 Visibility Parameter: Result and Analysis

The average of predicted visibility (Vis_{pred_avg}) was compared with the average of estimated visibility computed from the GPS collected data (Vis_{gps_avg}). To calculate Vis_{gps} at each GPS epoch, we classified active satellites into four groups based on Signal-to-Noise Ratio (SNR). According to the NMEA standard, SNR can range from 0 to 99 dB depending on how manufactures set the starting number. The two GPS receivers used in these experiments have the same SNR range, 0-51 dB. Thus, we divided the range into four groups with equal intervals, as defined in Table 4.1. Each group was mapped to the four predicted LOS types. Equation 4.5 was applied to compute Vis_{gps} with the same weights used for prediction. Note that this mapping may not truly reflect the actual LOSs perceived by the receiver but it is used to differentiate quality of received signals. In addition, GPS receivers may observe and use satellites at low elevation. Thus, for an unbiased comparison, only the satellites above the 10° elevation cutoff were included in Vis_{gps} calculation.

The statistics of Vis_{pred_avg} and Vis_{gps_avg} grouped by testing locations are summarized in Table 4.2. The number of GPS points for R1 was about half of the prediction epochs because R1 recorded GPS points every two seconds even though it was set for computing a position every second. R2 also had some missing epochs due to automatic sleep mode set by the manufacture.

Figure 4.22 shows the statistics of Vis_{pred_avg} and Vis_{gps_avg} computed from the 30 data sets. Open sky locations generally have high visibility ($Vis > 0.8$) with slight differences between prediction and GPS data. Overall, moderate and blocked locations have lower visibility.

Vis_{pred_avg} of moderate and blocked locations were significantly lower than Vis_{gps_avg}. As fewer satellites with direct LOS were observed at moderate and block locations, the visibility value dropped, in general $Vis < 0.8$. In addition, because the simulation has low sensitivity in predicting satellites at moderate and blocked locations, it may miss some satellites due to the multipath effect. Another factor that causes the differences is the use of high-sensitive receivers. The receivers used at least half of all satellites in view (both $Vis_{gps\_avg} > 0.5$), which increased positional availability but may not guarantee positional accuracy. Nevertheless, the predicted visibility with $Vis < 0.8$ can indicate potential obstructed areas. Further investigation is required for the use of combined GNSS constellations.

Table 4.1. Classification of GPS satellites based on the actual SNR used in Vis_{gps} calculation

Group	SNR	Mapping to LOS Types	Weight
SNR.1	SNR ≥ 38	Direct	1.0
SNR.2	25 ≤ SNR < 38	Diffraction	2/3
SNR.3	12 ≤ SNR < 25	Reflection	1/3
SNR.4	SNR < 12	Blocked	0.0

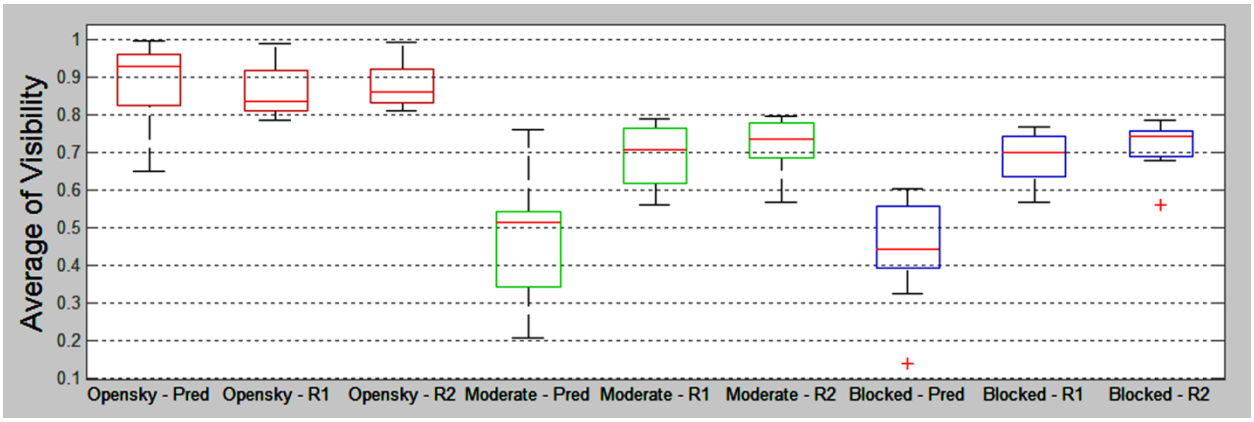


Figure 4.22. Average visibility

Table 4.2. Statistics of the numbers of different satellite types and the values of the visibility parameter

	Location	Prediction							R1							R2						
		#Points	# Total Sat.[5]	Average # of satellites with				Vis_{pred_avg}	#Points	# Total Sat.[5]	Average # of satellites with				Vis_{pred_avg}	#Points	# Total Sat.[5]	Average # of satellites with				Vis_{gps_avg}
				Dir.	Dif.	Ref.	Blk.				SNR.1[1]	SNR.2[2]	SNR.3[3]	SNR.4[4]				SNR.1[1]	SNR.2[2]	SNR.3[3]	SNR.4[4]	
Test 1	O1	900	9.0	7.0	1.7	0.0	0.3	0.905	450	9.0	6.0	1.7	0.5	0.8	0.811	597	9.0	6.6	1.3	0.2	1.0	0.832
	O2	900	10.0	9.2	0.8	0.0	0.0	0.972	450	10.0	9.0	1.0	0.0	0.0	0.967	900	10.0	9.6	0.4	0.0	0.0	0.983
	O3	900	10.0	7.5	2.2	0.0	0.3	0.898	443	10.0	5.6	3.5	0.6	0.3	0.814	806	9.9	6.9	1.7	0.2	1.1	0.820
	O4	900	9.2	9.1	0.1	0.0	0.0	0.997	450	9.4	9.1	0.3	0.0	0.0	0.990	604	9.6	9.3	0.2	0.0	0.0	0.991
	O5	900	9.0	6.4	1.3	0.0	1.2	0.813	450	9.0	6.0	1.3	0.7	1.0	0.785	900	9.0	6.7	1.7	0.2	0.4	0.878
Test 2	O1	900	9.3	7.6	0.1	0.0	1.6	0.824	450	9.6	7.4	0.4	0.3	1.5	0.809	900	9.3	7.5	0.4	0.2	1.2	0.845
	O2	900	8.3	4.6	1.2	0.0	2.5	0.650	450	8.3	5.6	1.5	0.0	1.1	0.797	883	7.9	5.6	1.4	0.0	0.8	0.834
	O3	900	8.0	7.3	0.6	0.0	0.2	0.956	450	8.0	6.0	1.7	0.1	0.3	0.890	891	8.0	7.0	0.5	0.0	0.4	0.922
	O4	900	8.6	7.4	1.3	0.0	0.0	0.951	450	8.7	5.9	2.3	0.1	0.4	0.859	900	8.7	5.6	2.2	0.1	0.8	0.812
	O5	900	9.0	8.1	0.7	0.0	0.1	0.959	450	9.0	7.8	0.8	0.0	0.5	0.919	888	9.1	7.7	0.7	0.0	0.6	0.908
Test 1	M1	900	9.0	5.0	0.4	0.8	2.8	0.619	441	9.0	5.1	2.5	0.1	1.2	0.763	651	9.0	4.9	3.4	0.1	0.6	0.796
	M2	900	8.8	3.7	1.6	0.0	3.5	0.543	450	8.9	2.5	3.6	1.5	1.2	0.616	836	8.8	3.1	3.9	0.7	1.1	0.672
	M3	900	9.4	0.8	1.5	0.4	6.7	0.209	450	9.6	4.5	3.3	0.4	1.4	0.708	900	9.2	5.0	3.1	0.2	0.9	0.777
	M4	900	8.0	3.0	1.1	0.8	3.1	0.502	425	8.0	3.5	2.8	0.7	1.0	0.702	900	8.0	3.7	2.7	0.5	1.1	0.707
	M5	900	9.0	2.9	1.1	0.0	4.9	0.410	450	9.0	2.7	2.9	1.1	2.2	0.560	598	9.0	2.2	3.8	1.2	1.8	0.569
Test 2	M1	900	9.1	4.1	1.0	0.0	4.0	0.527	450	9.4	4.5	2.7	0.8	1.4	0.699	888	9.0	4.1	3.0	0.6	1.4	0.693
	M2	900	9.0	3.8	1.6	0.0	3.6	0.544	450	9.0	2.2	3.1	2.6	1.2	0.568	890	8.9	3.6	3.3	0.7	1.2	0.686
	M3	900	8.7	2.1	1.0	0.0	5.6	0.318	450	8.8	4.4	3.4	0.3	0.7	0.774	892	8.8	4.7	3.0	0.1	0.9	0.770
	M4	900	9.0	3.0	0.1	0.0	5.9	0.344	450	9.0	3.9	3.7	0.5	0.9	0.730	866	9.0	4.8	3.0	0.3	0.9	0.765
	M5	900	9.0	5.4	2.1	0.0	1.4	0.760	450	9.1	5.8	1.8	0.4	1.1	0.788	881	9.1	5.5	2.2	0.4	0.9	0.784
Test 1	B1	900	8.5	1.0	0.2	0.0	7.3	0.142	450	8.6	1.6	3.9	1.8	1.2	0.568	900	8.8	1.7	4.1	1.5	1.5	0.562
	B2	900	7.1	2.1	1.8	0.0	3.2	0.486	446	7.7	2.8	1.8	1.7	1.4	0.600	600	7.0	3.2	2.0	0.5	1.3	0.679
	B3	900	5.5	2.5	1.2	0.0	1.8	0.602	445	5.9	2.3	2.1	1.0	0.4	0.692	727	5.7	2.4	2.2	0.4	0.8	0.694
	B4	900	7.4	2.0	0.4	0.2	4.7	0.326	450	7.4	2.1	3.3	1.0	1.0	0.634	900	7.5	2.4	3.6	0.7	0.7	0.687
	B5	900	8.9	2.9	1.2	0.4	4.5	0.425	446	9.1	4.0	3.0	1.1	1.0	0.703	615	8.6	4.6	2.2	1.1	0.6	0.757
Test 2	B1	900	8.2	3.0	0.3	0.0	4.9	0.391	450	8.3	4.9	2.0	0.3	1.1	0.768	900	8.3	5.0	2.0	0.1	1.1	0.776
	B2	900	8.0	2.0	2.3	0.0	3.6	0.451	435	8.0	2.3	3.4	1.8	0.5	0.652	893	8.0	3.3	3.6	0.8	0.3	0.743
	B3	900	9.3	5.0	0.2	0.1	4.0	0.557	450	9.4	5.5	1.9	0.8	1.1	0.756	900	9.1	5.5	1.9	0.9	0.7	0.787
	B4	900	9.4	3.2	1.2	0.1	4.8	0.435	450	9.4	4.0	3.8	0.4	1.3	0.704	900	9.5	4.3	4.0	0.1	1.0	0.741
	B5	900	8.0	3.9	1.1	0.0	3.0	0.581	450	8.0	3.7	3.2	0.4	0.7	0.743	887	8.0	3.9	2.8	0.5	0.8	0.744

[1] The number of satellites with SNR ≥ 38

[2] The number of satellites with $25 \leq$ SNR < 38

[3] The number of satellites with $12 \leq$ SNR < 25

[4] The number of satellites with SNR < 12

[5] The total number of satellite about 10° elevation cutoff

4.4.4.4 Availability Parameter: Result and Analysis

Figure 4.23 shows the statistics of the number of active satellites for all epochs. The number of active satellites measured by the two GPS receivers was always greater than four, (*Av* ≅ 1 for every epoch), which resulted in the maximum availability (100%) for all location types. For open sky locations, more than five satellites always were predicted, which ensured 100% positional availability at these locations. However, the moderate and blocked locations were sometimes predicted with less than four active satellites, which occasionally caused *Av* = 0 at some epochs. Thus, this made the overall positional availability of some obscured locations less than 100%. The locations with low average number of active satellites tend to have low availability. To ensure maximum availability of position solutions during a certain time period, at least the average of five active satellites is required.

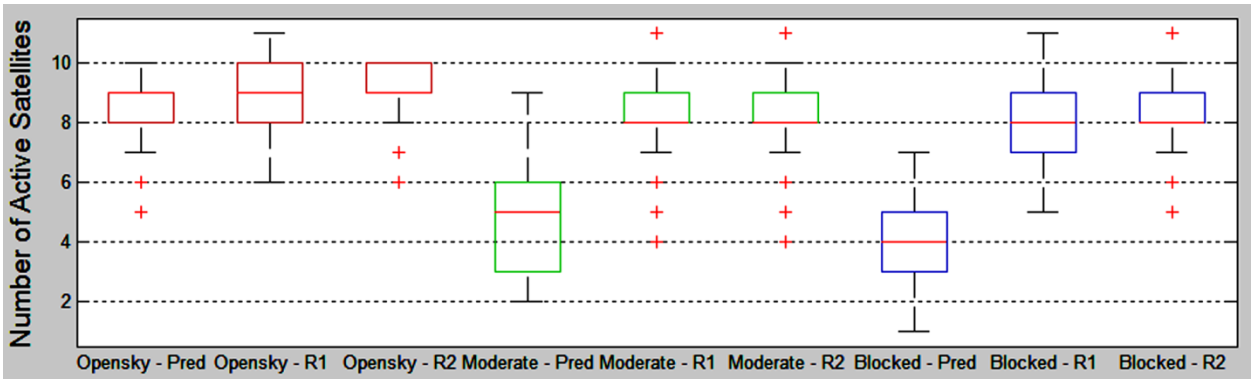
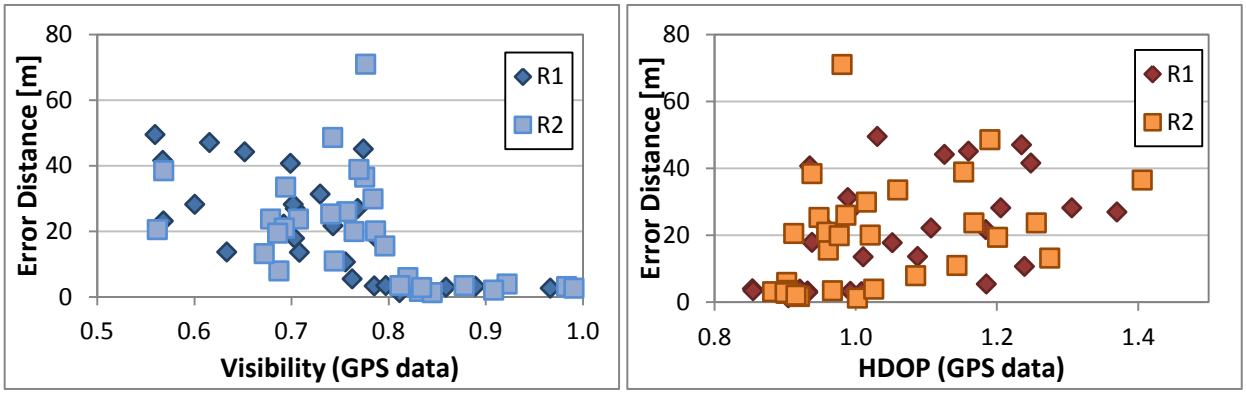


Figure 4.23. Number of active satellites from the prediction and GPS data

4.4.4.5 Accuracy Parameter: Result and Analysis

To emphasize that information outputted from GPS receivers may not reflect the actual level of positioning quality, Figure 4.24 shows the relationship between the average GPS error distances (in meters) and the average of (a) visibility (*Vis*) and (b) HDOP obtained from the two testing receivers at testing sites. These high-sensitive receivers always outputted optimistic values for visibility and HDOP even when positional accuracy was poor (distance errors > 10 m).

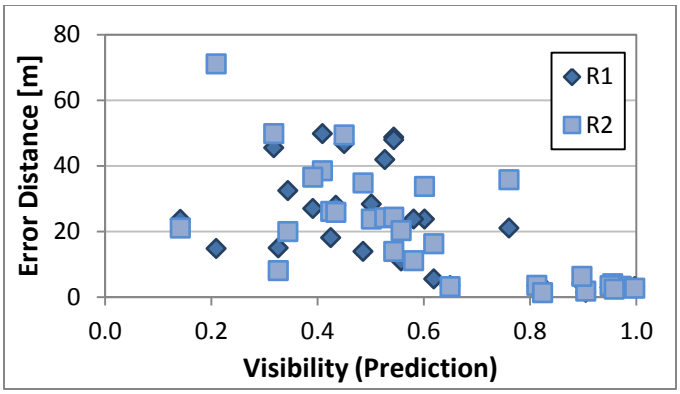
One reason is that they were able to acquire and use more satellites with weak signals. Thus, applications requiring high positioning accuracy (e.g., navigation) should not solely depend on HDOP estimated by GPS receivers.



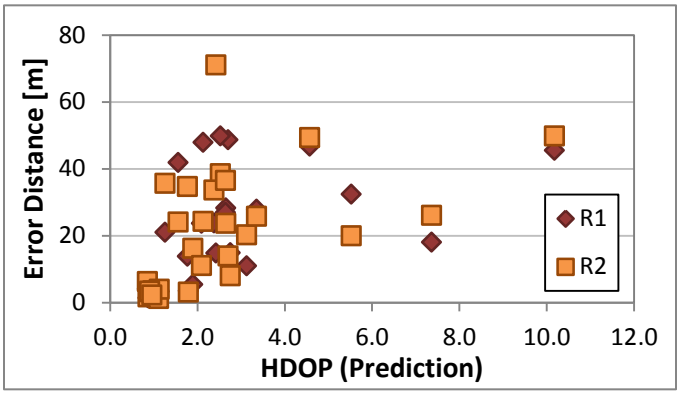
(a) (b)

Figure 4.24. Relationship between GPS error distances and the average (a) visibility and (b) HDOP of all testing locations obtained from the two GPS receivers

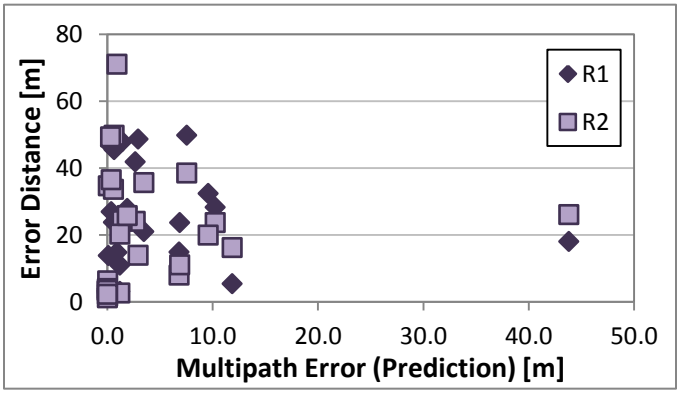
On the other hand, Figure 4.25 shows the relationships between the average error distances of the two GPS receivers and the average of the predicted (a) visibility, (b) HDOP, and (c) multipath error. The plots of visibility and HDOP show that under open sky condition the prediction values of these two variables are close to the actual GPS quality, while under moderate and blocked conditions the prediction values indicate fair and poor positional accuracy which is relatively lower than the actual GPS visibility and HDOP. The predicted multipath errors are minimal (≈ 0 m) under open sky condition but vary within a 0-50 m range for moderate and blocked conditions.



(a)



(b)



(c)

Figure 4.25. Relationships between GPS error distances and the average prediction values of (a) visibility, (b) HDOP, and (c) multipath error for all testing locations

The predicted accuracy (Ac_{pred}) at each testing location and epoch were compared against accuracy of the actual GPS data (Ac_{gps}). Ac_{gps} was measured by using an error distance (x)

between the GPS point and the reference coordinates, then the error distance was mapped to the accuracy range [0,1] using Equation 4.12.

$$GPS\ accuracy = \begin{cases} 1 - (x/50), & 0 < x \leq 50 \\ 0, & d > 0 \end{cases} \quad (4.12)$$

Table 4.3 summarizes the predicted values of visibility, HDOP, and multipath errors and accuracy of the GPS data at each testing location. These values were used to train the MFs, shown in Figure 4.15.

Figure 4.26 shows the statistics of Ac_{pred} computed at each epoch compared to Ac_{gps} of the two receivers under three environment settings. Under the open sky condition, accuracies obtained from prediction and GPS receivers were high with the medians of Ac_{pred} and Ac_{gps} 1.00 and 0.94, respectively. The differences were minor with 0.069 and 0.075 RMSE compared to R1 and R2 or about 3.45 and 3.75 m distance errors. This is due to the fact that the best predicted accuracy was quantified as 1.0 or 0 m error distance. However, it should be noted that the value of *Ac* is not intended to indicate distance errors but it gives a confidence level of obtaining position with high accuracy. Larger deviations of both Ac_{pred} and Ac_{gps} can be observed for moderate and blocked locations. Average Ac_{pred} for moderate and blocked is 0.386 and 0.326, respectively, with RMSE about 0.3 compared with the two receivers. The difference was mainly caused by the extrapolation process for estimating position estimation used in the receivers. Thus, this caused gradual changes in GPS distance errors over epochs.

Figure 4.27 shows the average of Ac_{pred} and Ac_{gps} at each test location with two different time periods. The average Ac_{gps} of the two receivers was almost the same in open sky locations with minor differences from Ac_{pred} (e.g., O1-1 vs. O1-2) but high discrepancies between the two receivers and the prediction were observed for moderate and block locations (e.g., M1-1 vs. M1-2 and B1-1 vs. B1-2). In addition, collecting GPS data at different time of the day may cause differences on level of positional accuracy.

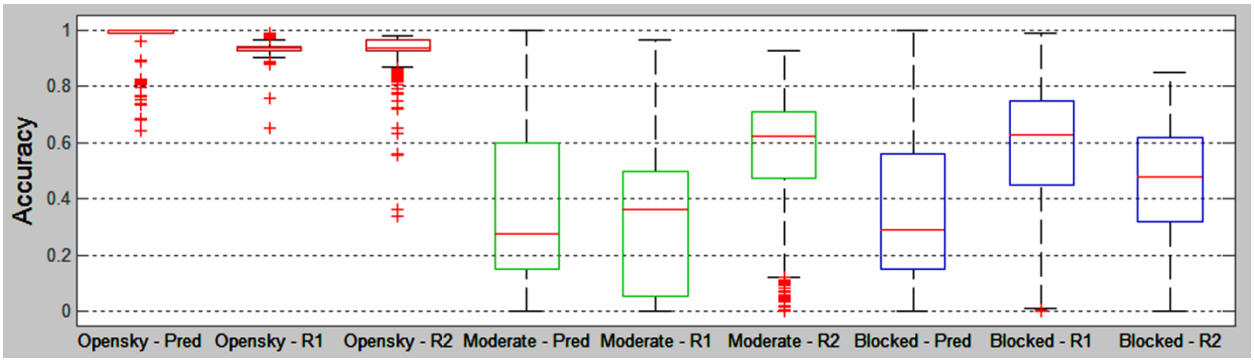


Figure 4.26. Predicted accuracy and GPS accuracy

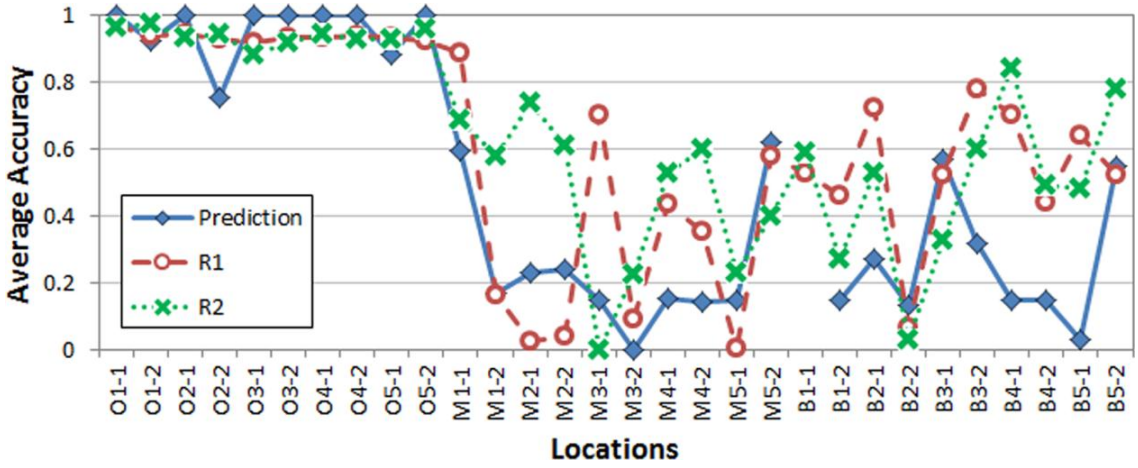


Figure 4.27. Average accuracy at each testing location

Table 4.3. Statistics of prediction and GPS data used for training the MFs of the fuzzy logic model

	Location	Date	Time Duration		Statistics of Prediction Data			Statistics of GPS data			
					Average of			RMS distance errors [m]		Ac_{gps}[a]	
			From	To	*Vis*[a]	*HDOP*[a]	*ME*[a] [m]	R1	R2	R1	R2
Test 1	O1-1	08/30/10	23:20:00	23:34:59	0.905	0.867	0.037	1.403	1.743	0.972	0.965
	O2-1	09/05/10	0:30:00	0:44:59	0.972	0.891	0.005	2.692	3.203	0.946	0.936
	O3-1	09/05/10	0:01:38	0:16:37	0.898	0.851	0.037	4.063	6.252	0.919	0.875
	O4-1	09/05/10	0:56:00	1:10:59	0.997	0.977	1.197	3.154	2.644	0.937	0.947
	O5-1	09/04/10	23:11:00	23:25:59	0.813	0.980	0.002	3.386	3.498	0.932	0.930
Test 2	O1-2	09/14/10	17:33:00	17:47:59	0.824	1.106	0.001	3.093	1.272	0.938	0.975
	O2-2	09/26/10	18:35:00	18:49:59	0.650	1.795	0.036	3.477	3.102	0.930	0.938
	O3-2	09/18/10	18:42:30	18:57:29	0.956	1.126	0.006	3.299	4.001	0.934	0.920
	O4-2	09/26/10	18:59:00	19:13:59	0.951	0.902	0.012	2.939	3.461	0.941	0.931
	O5-2	09/18/10	17:27:00	17:41:59	0.959	0.953	0.016	3.878	2.320	0.922	0.954
Test 1	M1-1	08/28/10	22:14:35	22:29:35	0.619	1.891	11.850	5.460	16.212	0.891	0.676
	M2-1	08/28/10	22:40:00	22:54:59	0.543	2.701	2.934	48.681	13.908	0.026	0.722
	M3-1	08/30/10	22:56:29	23:11:29	0.209	2.424	0.906	14.792	71.043	0.704	0.000
	M4-1	08/31/10	19:30:00	19:44:59	0.502	2.652	10.237	28.310	23.700	0.434	0.526
	M5-1	09/01/10	23:03:35	23:18:34	0.410	2.527	7.552	49.809	38.482	0.004	0.230
Test 2	M1-2	10/01/10	20:36:00	20:50:59	0.527	1.559	2.634	41.848	24.067	0.163	0.519
	M2-2	10/01/10	20:16:00	20:30:59	0.544	2.126	1.389	47.927	24.263	0.041	0.515
	M3-2	09/14/10	17:59:00	18:13:59	0.318	10.180	0.634	45.506	49.788	0.090	0.004
	M4-2	09/14/10	17:06:15	17:21:14	0.344	5.525	9.564	32.377	19.922	0.352	0.602
	M5-2	09/18/10	16:51:00	17:05:59	0.760	1.256	3.458	20.998	35.675	0.580	0.286
Test 1	B1-1	09/01/10	23:30:00	23:44:59	0.142	N/A	N/A	23.617	20.974	0.528	0.581
	B2-1	09/01/10	23:56:18	0:11:17	0.486	1.767	0.080	13.825	34.710	0.723	0.306
	B3-1	09/04/10	23:36:00	23:50:59	0.602	2.376	0.569	23.766	33.646	0.525	0.327
	B4-1	08/31/10	19:50:00	20:04:59	0.326	2.754	6.797	14.942	7.969	0.701	0.841
	B5-1	09/01/10	22:40:18	22:55:17	0.425	7.366	43.820	18.015	26.044	0.640	0.479
Test 2	B1-2	09/18/10	17:50:00	18:04:59	0.391	2.641	0.375	26.958	36.470	0.461	0.271
	B2-2	09/18/10	18:10:00	18:24:59	0.451	4.567	0.299	46.772	49.327	0.065	0.013
	B3-2	09/18/10	16:32:00	16:46:59	0.557	3.129	1.174	10.955	20.246	0.781	0.595
	B4-2	09/14/10	16:48:00	17:02:59	0.435	3.350	1.898	27.882	25.752	0.442	0.485
	B5-2	09/26/10	18:10:00	18:24:59	0.581	2.091	6.857	23.728	11.084	0.525	0.778

[a] Values used for training the MFs in Figure 4.15

5.0 IGNSS QOS PREDICTION IN NAVIGATION

5.1 INTRODUCTION

This chapter provides the details of the other three modules of iGNSS QoS prediction described in Chapter 3, which are segment sampling, tracking-based iGNSS QoS prediction, and iGNSS QoS segmentation. Given a route segment as input, the segment sampling module takes sample points on the segment, and passes them on to the point-based prediction module described in Chapter 4. The tracking-based prediction module uses the information from the point-based prediction to estimate four iGNSS QoS parameters on each segment: average availability ($\widetilde{Av}$), average accuracy ($\widetilde{Ac}$), continuity (*Co*), and reliability (*Re*). Different levels of QoS are identified on each segment through the iGNSS QoS segmentation module.

To evaluate the output of track-based prediction and iGNSS QoS segmentation, an experiment to compare the prediction against the QoS of real GPS trajectories was conducted. Factors impacting the prediction explored in this experiment are sampling distance, 3D data granularity, prediction time, minimum accuracy requirement, and chunk's QoS variation tolerance.

5.2 SEGMENT SAMPLING

In this dissertation, the systematic sampling method with fixed-distance interval was chosen Figure 5.1 illustrates points on a route that are selected by the segment sampling module with a fixed-distance interval. A route (*R*) is defined as a path from a start location (origin) to an end location (destination). Let *R* denote a series of *n* connected route segments, $R = \{L_1, L_2, L_3, \dots, L_n\}$ and L_i is the i^{th} segment of the route. Each segment has two decision

nodes, one incoming and another outgoing; each located at the edge of the segment. Each segment composes of a piecewise-linear line or multiple piecewise-linear lines, specified by a sequence of (shape) points to represent, as close as possible, the actual shape of the road. L_2 in Figure 5.1 has p_5 as an incoming (decision) node, p_{13} as an outgoing (decision) node, and p_7, p_9, p_{11} as shape points. On each segment, intermediate points are selected with a fixed distance interval (d) from the incoming node or a shape point to the next shape point or the outgoing node until the distance between the intermediate point and the next point is $\leq d$. Given $\|\langle p_a, p_b \rangle\|$ is a piecewise distance between p_a and p_b, Figure 5.1 shows that L_2 has $\|\langle p_9, p_{10} \rangle\| = \|\langle p_{11}, p_{12} \rangle\| = d$ and $\|\langle p_{10}, p_{11} \rangle\|, \|\langle p_{12}, p_{13} \rangle\| \leq d$. The segment sampling module outputs a set of sample points $P = \{p_1, p_2, p_3, \dots, p_m\}$ where m is the total number of sample points including decision nodes, shape points, and intermediate points. Given a time when the user starts to travel, the time at which the user would pass each point could be estimated by using travel speed or posted speed limit on each segment.

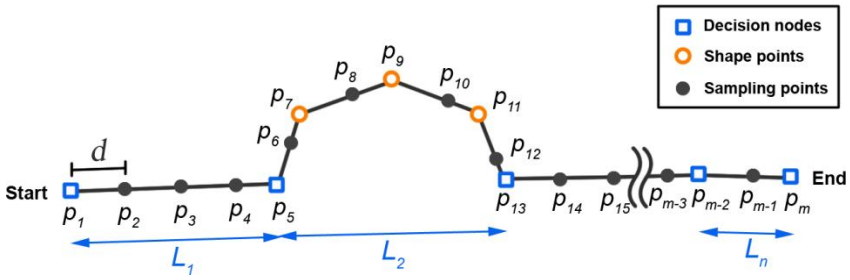


Figure 5.1. Points selected by the segment sampling module on a given route

5.3 TRACKING-BASED iGNSS QoS PARAMETERS

Let a segment L_i contain of a sequence of k sample points in a spatial order from an incoming node to an outgoing node $\{p_1, p_2, \dots, p_k\}$ on the i^{th} segment. It is assumed that each point (p_j) on L_i is a representative of the sub-segment (l_j), as depicted in Figure 5.2. The length of each sub-segment is a weight in the weighted average functions for calculating average availability and accuracy, $w_j = \|l_j\|$ for p_j, as:

$$w_j = \begin{cases} \left\| \langle p_j, \frac{p_j + p_{j+1}}{2} \rangle \right\| & ; j = 1 \\ \left\| \langle \frac{p_{j-1} + p_j}{2}, p_j \rangle \right\| + \left\| \langle p_j, \frac{p_j + p_{j+1}}{2} \rangle \right\| & ; 1 < j < k \\ \left\| \langle \frac{p_{j-1} + p_j}{2}, p_j \rangle \right\| & ; j = k \end{cases} \quad (5.1)$$

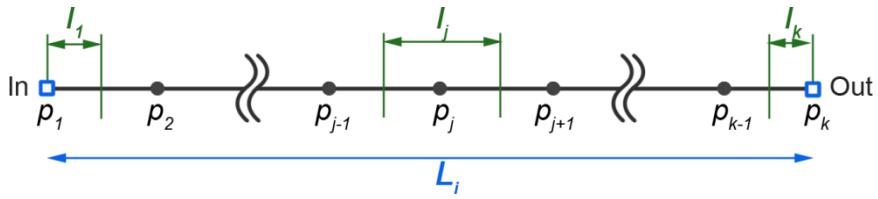


Figure 5.2. Sub-segments of a route segment used in weight calculation

Average availability ($\widetilde{Av}$) is defined as a weighted average of availability of positioning solutions on a segment. It is calculated as follows:

$$\widetilde{Av}(L_i) = \frac{\sum_{j=1}^{k} w_j \cdot Av(p_j)}{\sum_{j=1}^{k} w_j}, \quad (5.2)$$

where $Av(p_j)$ is the iGNSS QoS availability at p_j and w_j is the length of the sub-segment l_j for p_j. $Av(p_j) = 1$ if a positioning solution is available, otherwise $Av(p_j) = 0$. $\sum_{j=1}^{k} w_j$ is always equal to the segment length.

Average accuracy ($\widetilde{Ac}$) is defined as a weighted average of positional accuracy on a segment, which is calculated as follows:

$$\widetilde{Ac}(L_i) = \frac{\sum_j w_j \cdot Ac(p_j)}{\sum_j w_j} \quad (5.3)$$

with a condition that $Av(p_j) = 1$, otherwise discard j^{th}. $Ac(p_j)$ is the iGNSS QoS accuracy at p_j, $0 \leq Ac(p_j) \leq 1$ and w_j is the length of the sub-segment l_j for p_j. It is possible that $\sum_j w$ is less than the segment length. Thus, $\widetilde{Ac}$ is an average accuracy of the available positioning solutions.

Continuity (*Co*) is the maximum probability that positioning solutions with a minimum accuracy can continuously be maintained without disruption on a given segment. Continuity is measured by the ratio of the longest length of the consecutive sub-segments that meet a minimum availability and accuracy to the total length of the segment, expressed as:

$$Co(L_i) = \frac{\max(\sum_{s=a}^{b} w_s)}{\sum_{j=1}^{k} w_j}, \tag{5.4}$$

where $\sum_{s=a}^{b} w_s$ is the sum of the length of consecutive sub-segments (from the a^{th} to b^{th} sub-segment) whose $Av(p_s) = 1$ and $Ac(p_s) \geq \sigma_{\text{threshold}}$, $\sigma_{\text{threshold}}$ is the minimum required accuracy set by navigation applications. *Co* = 1 means that iGNSS QoS will be continuously maintained while *Co* = 0 means there is a high probability that there will be no continuity of iGNSS QoS on the segment.

Reliability (*Re*) is the probability of obtaining positioning solutions with a minimum accuracy on a given segment. Reliability is the ratio of the sum of the sub-segment lengths that have availability and accuracy above the minimum requirements to the total length of the segment, expressed as:

$$Re(L_i) = \frac{\sum_{r} w_r}{\sum_{j=1}^{k} w_j}. \tag{5.5}$$

where $\sum_{r} w_r$ is the total length of sub-segments whose $Av(p_r) = 1$ and $Ac(p_r) \geq \sigma_{\text{threshold}}$. *Re* = 1 means that iGNSS solutions meet the application's requirement while *Re* = 0 means that there is a high probability of unreliable iGNSS solutions on the segment.

To clarify the calculation of each tracking-based iGNSS QoS parameter, an example is shown in Figure 5.3 where $Av(p_j)$ and $Ac(p_j)$ on the segment L_1 are given.

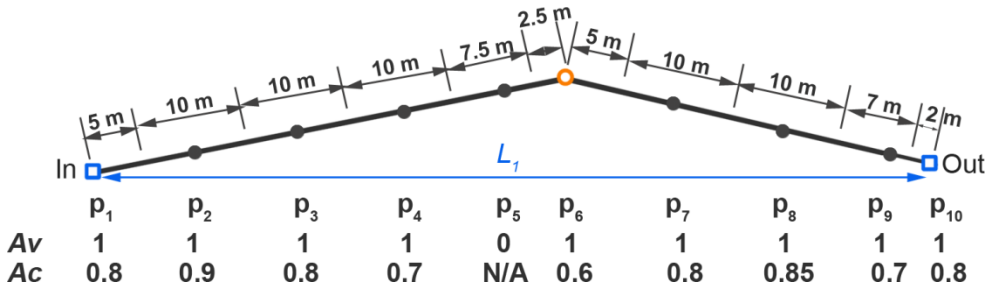


Figure 5.3. Example of navigation-based iGNSS QoS parameters

Assuming that a navigation application requires $\sigma_{\text{threshold}} = 0.8$, the following are values of the four parameters on L_1:

$$\widetilde{Av} = \frac{5+10+10+10+7.5+10+10+7+2}{5+10+10+10+7.5+7.5+10+10+7+2} = \frac{71.5}{79} = 0.905,$$

$$\widetilde{Ac} = \frac{5(0.8)+10(0.9)+10(0.8)+10(0.7)+7.5(0.6)+10(0.8)+10(0.85)+7(0.7)+2(0.8)}{5+10+10+10+7.5+10+10+7+2} = \frac{55.5}{71.5} = 0.776,$$

$$Co = \frac{\max(5+10+10,10+10,2)}{5+10+10+10+7.5+7.5+10+10+7+2} = \frac{25}{79} = 0.316,$$

$$Re = \frac{5+10+10+10+10+2}{5+10+10+10+7.5+7.5+10+10+7+2} = \frac{47}{79} = 0.595.$$

This shows that on average this segment has high availability and accuracy of positioning solutions but rather low solution continuity. There is about 59.5% chance that the solutions will meet the accuracy requirement.

5.4 IGNSS QOS SEGMENTATION

iGNSS QoS segmentation has two main steps: (1) merging and (2) chunking. In the first step, a *Sequence Merging Algorithm* (SMA) is developed to assign point-based QoS into sequences of points based on geometry and iGNSS QoS variation. This algorithm also considers points as a representative of sub-segments as described in Section 5.3 and Figure 5.2. Let segment L have k points $L = \{p_1, p_2, \dots, p_k\}$, in a spatial order from an incoming node to an outgoing node, which results in corresponding k sub-segments. Each point contains an attribute used in the merging process. For iGNSS QoS prediction, the predicted accuracy at each point, $Ac(p_j)$, is used as the attribute value in SMA. This is because positional accuracy is the main factor that highly impacts navigation performances. The objective of SMA is to merge these points on a segment into h sequences with minimal attribute variation and no spatial overlap. Thus, the points on L will be partitioned into $C = \{C_1, \dots, C_h\}$ where $0 < h \leq k$, C_i is the i^{th} sequence follows C_{i-1} and is followed by C_{i+1}. C_i contains $\langle p_{start[i]}, \dots, p_{end[i]} \rangle$ and C_{i+1} contains $\langle p_{start[i+1]}, \dots, p_{end[i+1]} \rangle$ where $p_{start[i+1]}$ follows $p_{end[i]}$; *start* and *end* are the index vector of the starting and ending point for each sequence.

An outline of SMA is given in Figure 5.4. The algorithm starts by setting each point as an individual sequence. Thus, the initial process begins with k sequences. To find an adjacent

sequence that its variation of attribute values is minimum, a weighted standard deviation (*S*) is defined as a merging function. *S* is computed from two sets of attribute values belonging to a pair of adjacent sequences. Thus, the initial process has *k*-1 pairs. *S* can be computed by:

$$S_i = S_{C_i,C_{i+1}} = \sqrt{\frac{\sum_{j=start[i]}^{end[i+1]} w_j}{\left(\sum_{j=start[i]}^{end[i+1]} w_j\right)^2 - \sum_{j=\text{start[i]}}^{end[i+1]} {w_j}^2} \cdot \sum_{j=start[i]}^{end[i+1]} \left(w_j \cdot \left(Ac(p_j) - \widetilde{Ac}_{C_i,C_{i+1}}\right)^2\right)} \quad (5.6)$$

$$\widetilde{Ac}_{C_i,C_{i+1}} = \frac{\sum_{j=start[i]}^{end[i+1]} w_j \cdot Ac(p_j)}{\sum_{j=start[i]}^{end[i+1]} w_j} \quad (5.7)$$

where $w_j$is the length of the sub-segment represented by p_j, given by Equation 5.1.

The algorithm merges the adjacent sequences pair that has the minimum weighted standard deviation, $\hat{S}$:

$$\hat{S} = \arg\min_i \{S_i\} \quad (5.8)$$

where $i = 1,2, \dots, |C|$. $|C|$ is the number of sequences.

In each iteration, all index vectors are updated according to new sequences. To reduce the computational load, only *S* of those sequences that are adjacent to the merged sequence are re-calculated. The process is repeated until the number of sequences is 1 or $\hat{S} > \tau$ where τ is the variation tolerance within each sequence.

It should be noted that SMA is not a clustering algorithm and differs from existing sequence clustering algorithms used in bioinformatics (Enright and Ouzounis, 2000, Krause et al., 2000) and web mining (Kumar et al., 2007, Park et al., 2008) in several respects. First, the concept of sequence merging is different from the concept of sequence clustering in that SMA takes into account attribute similarity within each sequence and topology of items (i.e., points or sub-segments). Second, unlike other sequence clustering algorithms that generally require an initial set of sample sequences in order to search for a sequence, SMA does not require a specific number of segments. Third, unlike traditional clustering algorithms (e.g., hierarchical clustering), SMA does not require an initial number of sequences.

The second step of the iGNSS QoS segmentation module is to construct line geometry of chunks based on the sequences assigned by SMA and to provide a statistical summary of iGNSS

QoS using the weighted average accuracy ($\widetilde{Ac}$ on chunks). In case of $|C| > 1$, chunk polylines can be reconstructed by:

$$C_i = \begin{cases} \left\{p_i, \dots, p_{shape\ points}, \dots, \frac{p_{end[i]}+p_{start[i+1]}}{2}\right\} & ;\ i = 1 \\ \left\{\frac{p_{end[i-1]}+p_{start[i]}}{2}, \dots, p_{shape\ points}, \dots, p_i\right\} & ;\ i = |C| \\ \left\{\frac{p_{end[i-1]}+p_{start[i]}}{2}, \dots, p_{shape\ points}, \dots, \frac{p_{end[i]}+p_{start[i+1]}}{2}\right\} & ;\ 1 < j < k \end{cases} \quad (5.9)$$

where $p_{shape\ points}$ are original shape points of the segment if exist.

In case of $|C| = 1$, the original line segment shape will be returned, which is given by:

$$C_1 = \{p_1, p_{shape\ points}, p_{end[1]}\}. \quad (5.10)$$

$\widetilde{Ac}(C_i)$ can be computed by Equation 5.3 to inform positional accuracy on chunks.

Input parameters:

P_L = A set of k points on a segment = $\{p_1, p_2, \dots, p_k\}$

A_L = A set of attribute values for each point in P_L

τ = Variation tolerance within sequences

Output parameters:

C = A set of sequences

SMA(P_L, A_L, τ){

Set points to individual sequences

Compute a distance weight, w_j, for each point using Equation 5.1

Repeat

for all pairs of adjacent sequences **do** {

Compute a weighted standard deviation, S, using Equation 5.6 - 5.7

}

Merge the adjacent sequences pair that has the minimum S ($\hat{S}$), if $\hat{S} \leq \tau$

Until $|C|$=1 or $\hat{S} > \tau$

}

Figure 5.4. Sequence merging algorithm

Figure 5.5 shows an example of iGNSS QoS segmentation with τ = 0.1. Note that the actual $Ac(p_2)$ is null (N/A) because of predicted position unavailability but it is assigned $Ac(p_2)$ = -10 for calculation purpose.

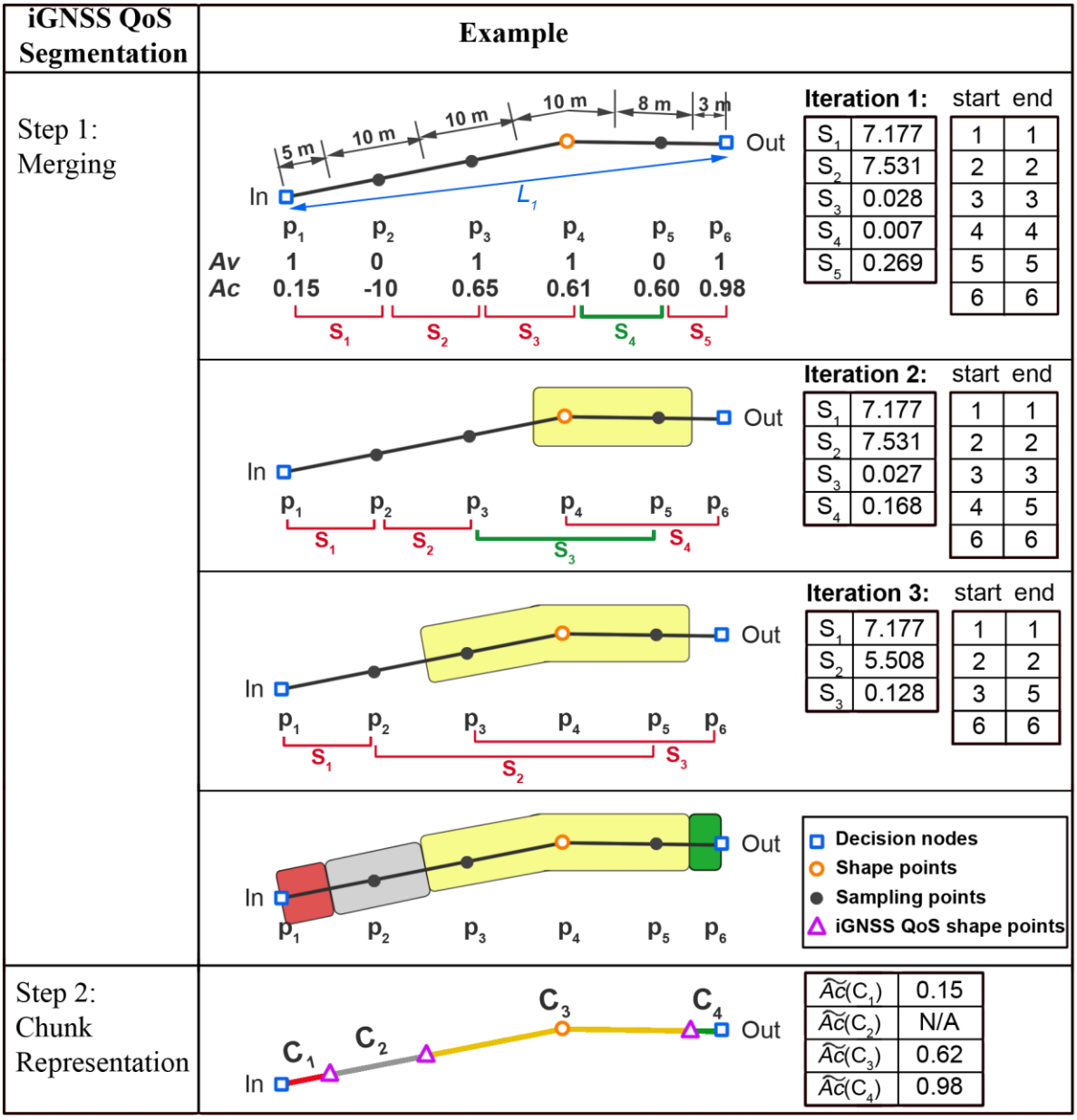


Figure 5.5. Example of iGNSS QoS segmentation (τ = 0.1)

5.5 EVALUATION

To evaluate the iGNSS QoS prediction methodology, the prediction results were compared against actual GPS baselines collected on a set of testing routes. Only GPS QoS was evaluated as currently GPS is the only GNSS in full operation.

5.5.1 Study Area and Testing Routes

The study area covers the downtown area of Pittsburgh and the University of Pittsburgh's main campus. iGNSS QoS prediction was tested with two modes of travel: driving and walking. Testing routes are pre-planned given origin and destination (O-D) pairs.

The driving routes were obtained from the Google Directions API[6], which are the first-returned route for each O-D pair. Routes provided by Google Maps are not necessarily shortest routes; our analysis of the routes produced by Google Maps indicates that other criteria, which are not revealed to the user, are used in finding routes (e.g., least turns). Three categories of driving routes were identified based on surrounding environments: open sky (O), high-rise building (H), and mixed environment (M). Four O-D pairs were chosen for each route category, as shown in Figure 5.6. For each route, GPS data were collected at three different times.

Walking routes were computed using a sidewalk network and Dijkstra's algorithm. The sidewalk network covers the University of Pittsburgh's main campus. It was collected by digitization of an orthoimage file with 1-m resolution and field survey using a DGPS receiver (Kasemsuppakorn and Karimi, 2009b). Due to the limited extent of the sidewalk network, only walking routes in mixed environment areas were identified. Five O-D pairs were chosen, as shown in Figure 5.7. Each walking route was visited at three different times for GPS data collection. Table 5.1 summarizes the characteristics of the testing routes.

[6] http://code.google.com/apis/maps/documentation/directions/

Table 5.1. Characteristics of testing routes

	Driving	**Walking**
Study area	37.2 km^2	1.8 km^2
Route categories based on environment settings	3 categories • Open sky route (O) • High-rise building route (H) • Mixed environment route (M)	1 category • Mixed environment route
Number of testing routes per environment setting	4 routes	5 routes
Repeated measurements	3 different times of travel: 12 pm to 0 am UTC (from 3/14/2011 to 3/20/2011)	3 different times of travel: 12 pm to 0 am UTC (from 3/14/2011 to 3/20/2011)
Total trajectories	36	15
Average distance of chosen routes [min | max]	O routes: 2.7 km [2.3 | 3.5] H routes: 1.2 km [1.0 | 1.4] M routes: 5.3 km [4.5 | 5.8]	1.0 km [0.7 | 1.1]

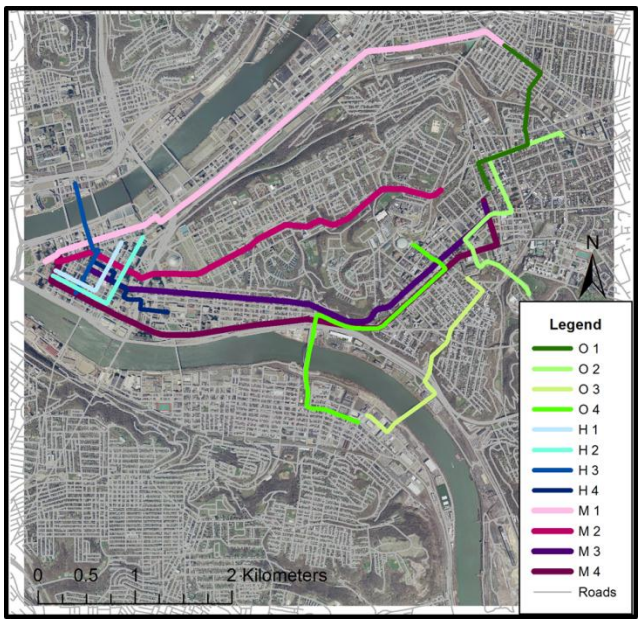


Figure 5.6. Driving routes

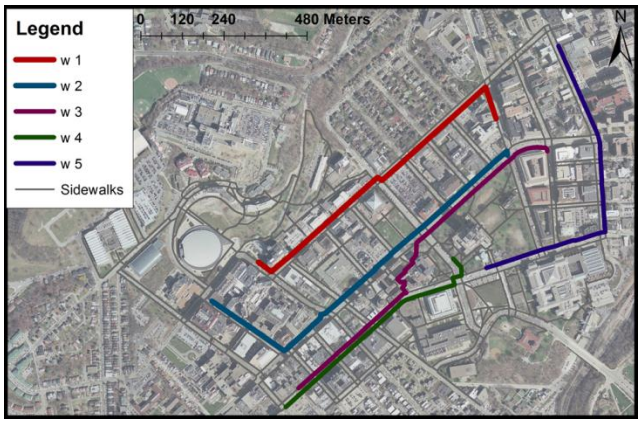


Figure 5.7. Walking routes

5.5.2 GPS Receivers and Data Collection

Two GPS receivers with E-TEK EB85A and MTK II chipset, referred to as R1 and R2, were used as described in Section 4.4.2. Both receivers were set to compute a position every second. The two receives were placed next to each other while collecting data. For driving they were placed on the middle of the car dashboard and for walking they were placed on the data collector's palm at shoulder height (about 1.2 m) with the horizontal direction parallel to the ground.

5.5.3 Method

The evaluation was divided into two analyses. Analysis I was aimed to evaluate tracking-based iGNSS QoS prediction and Analysis II was aimed to assess iGNSS QoS segmentation. The steps for preparing GPS baseline used in both analyses are summarized in Table 5.2. Table 5.3 summarizes variables used in both analyses.

Table 5.2. Steps for preparing GPS baseline used in iGNSS QoS prediction evaluation

Steps for GPS baseline preparation	Description
Step 1: Point collection	1.1 Collect GPS data along planned route
Step 2: Point-based QoS	2.1 Digitize trajectory on an orthoimage (reference polyline) 2.2 Snap decision nodes of the planned route on the reference polyline and divide the reference polyline into segments according to the snapped decision nodes (reference segments) 2.3 Map match each GPS point on the correct reference segment using visual inspection and perpendicular distance to the segment 2.4 Compute availability (*Av*) and accuracy (*Ac*) for each map-matched point as follows: • *Av* is validity of 3D point and existence of solutions based on time stamp • *Ac* is computed by $1 - x/50$ where *x* is distance between GPS and the map-matched point. If $x > 50$, $Ac = 0$
Step 3: Tracking-based QoS	3.1 Compute the tracking-based QoS parameters from GPS *Av* and *Ac* using Equation 5.1-5.5
Step 4: QoS segmentation	4.1 Assign map-matched GPS points obtained from Step 2.4 to corresponding prediction chunks where points are located 4.2 Compute average accuracy ($\widetilde{Ac}$) on each chunk based on the point-based QoS using Equation 5.3

In Analysis I, the four tracking-based QoS parameters, i.e., $\widetilde{Av}$, $\widetilde{Ac}$, *Co*, and *Re*, were analyzed. For each route type, differences between the prediction results and GPS baselines were measured by a distance-weighted RMSE which takes into account the distance of route segments

as a normalization factor; thus each route has the total weight equal to 1. Friedman's two-way ANalysis of VAriance (ANOVA) of ranks, due to the non-normal distribution of the data, was used to compare statistical differences among the prediction groups when manipulating the experimenting variables.

In Analysis II, $\widetilde{Ac}$ on chunks, chunk lengths, and number of chunks were measured. For each route type, a distance-weighted RMSE of $\widetilde{Ac}$ on chunks was computed where weight is the chunk length and each route has a total weight equal to 1. The Kruskal-Wallis test, a one-way ANOVA by ranks, due to the non-normal distribution and non-equal size of the prediction data among groups, was used to compare statistical differences among the prediction groups when manipulating the experimenting variables.

Table 5.3. Variables used for experimentation with iGNSS QoS prediction

Experimentation Variables	**Drive**	**Walk**
Different time of travel	3	
Sampling distance [m]	20, 40, 60, 80, and 100	10, 20, 30, 40, and 50
Minimum accuracy requirement, $\sigma_{threshold}$ (Analysis I)	0.2, 0.4, 0.6, and 0.8	
Variation tolerance within each chunk, τ (Analysis II)	0.1, 0.2, 0.3, 0.4, and 0.5	
Granularities of TINs used in point-based prediction	3-zone, 2-zone and 1-zone LOS calculation	

5.5.4 Results and Analyses

Table 5.4 summarizes the segment length of the testing routes and the travelling speed used during data collection. As driving routes from Google's Direction API were divided into segments based on directional changes at decision nodes, we observed that Google's driving routes have a small number of segments but high variations in segment lengths, in particular mixed routes which include short local roads and long free-way roads. On the other hand,

walking routes computed from the sidewalk network have a large number of segments. For driving mode, GPS data were collected according to the flow of traffic. The average speed on high-rise building routes in urban areas was low due to frequent stops at multiple traffic lights. For walking mode, the data collection was set at a steady speed on sidewalks.

Table 5.4. Statistics of segment lengths and travelling speeds on segments

Route Types	*N*[1]	Segment Length [m]			GPS Speed per Segment[2] [m/s]		
		Min.	Median	Max.	Min.	Median	Max.
Driving							
• Open sky (O)	72	4.06	471.09	1,589.56	5.52	14.53	27.86
• High-rise building (H)	45	65.43	249.99	683.36	2.29	10.67	36.06
• Mixed environment (M)	45	5.62	661.45	4,302.14	7.25	15.83	17.93
Walking	369	1.32	24.11	280.69	0.44	2.41	8.17

[1] Number of route segments; [2] Average of R1 and R2

5.5.4.1 Analysis I: Tracking-Based iGNSS QoS

The prediction results and GPS baselines of the tracking-based parameters are reported in Figures 5.8 – 5.11 for driving routes and in Figures 5.12 – 5.15 for walking routes.

Sampling Distance

Figures 5.8 and 5.12 show the prediction results and GPS baselines with various sampling distances. Overall, sampling distance has a minor impact on the difference between the prediction and the baselines. Large RMSEs can be observed for the routes under obstructed environments and small RMSEs for open sky routes, as shown by RMSEs in Figure 5.11 for driving and Figure 5.15 for walking. On average, the prediction values of the four parameters were relatively lower than the GPS baselines, as reported in Table 5.5. There are several reasons for this. First, the GPS receivers are highly sensitive to signals, which they can lock onto satellite signals and maintain accuracy even operated in difficult surrounding environments. Second, modern GPS receivers, including the ones used in this study, supports filters (e.g., Kalman filter)

to reduce measurement noises and improve accuracy of position estimation for kinematic mode of data collection. Thus, accuracies of GPS baselines were high and varied within a small range; Figure 5.8(b) shows the GPS baseline for driving and Figure 5.12(b) shows the GPS baseline for walking. Third, the current tracking-based prediction calculation does not take into account dynamism of the user (i.e., speed and direction). The prediction was based only on the results of the point-based prediction module in Chapter 4, which were tuned by the data collected in static mode.

To evaluate the differences among the prediction results with various sampling distances, the Friedman tests revealed no significant differences on $\widetilde{Av}$ and $\widetilde{Ac}$ for all route types ($p > 0.05$). Indifferences of $\widetilde{Av}$ and $\widetilde{Ac}$ with various sampling distances were due to the weighted average function defined for both parameters. This is especially true for long segments where the sampling distance is much less than the segment length ($d \ll \|L\|$). The significant differences were found on *Re* and *Co* of the mixed environment driving routes ($\chi^2(44) = 12.68$ and 31.45, $p < 0.05$) and *Co* of the walking routes ($\chi^2(368) = 12.31$, $p < 0.05$). *Co* increased as the sampling size increased.

Table 5.5. Statistics of tracking-based iGNSS QoS prediction (using 3-zone LOS calculation), GPS QoS, and RMSE

			Prediction (Median)				GPS[2] (Median)				RMSE			
Travel Mode	Type	N[1]	$\widetilde{Av}$	$\widetilde{Ac}$	*Re*	*Co*	$\widetilde{Av}$	$\widetilde{Ac}$	*Re*	*Co*	$\widetilde{Av}$	$\widetilde{Ac}$	*Re*	*Co*
Driving	O	72	1.00	0.82	1.00	1.00	1.00	0.95	1.00	1.00	0.06	0.20	0.19	0.25
(sampling = 60	H	45	0.50	0.29	0.04	0.04	1.00	0.82	0.98	0.96	0.55	0.48	0.73	0.69
m, σ = 0.4)	M	45	1.00	0.70	0.86	0.52	1.00	0.90	1.00	1.00	0.15	0.24	0.32	0.34
Walking (sampling = 30 m, σ = 0.6)	M	369	1.00	0.70	0.74	0.73	1.00	0.92	1.00	1.00	0.20	0.34	0.48	0.49

[1] Number of route segments; [2] Average of R1 and R2

Considering that sampling distance is a tradeoff between computation time and correctness of the results, the smaller the sampling distance, the more number of sample points is generated. With a large number of sample points, more computation resources are required for

predicting point-based iGNSS QoS. As of the current implementation on a computer with 2.13-GHz dual core processors with 2GB of RAM, it took about 33.28, 22.38, and 4.37 seconds to predict a point-based iGNSS QoS when using 3-zone LOS calculation, 2-zone LOS calculation, and 1-zone LOS calculation, respectively, as illustrated in Figure 4.2. Thus, instead of applying a sampling distance to all areas, a dynamic schema for choosing a sampling distance would be more appropriate. The dynamic sampling scheme, or stratified sampling method, should adapt based on segment lengths, changes in surrounding environments, or changes based on prior knowledge of iGNSS QoS to minimize computation time and maximize correctness of the prediction results.

TIN Data Layers

Granularity of TIN data layers used in point-based prediction is another factor affecting computation time and correctness of the results. The 3-zone LOS calculation provides fine details of the 3D models and yields high precision of prediction but it involves a large amount of data imposing high computation time. As computation time in real-time navigation applications is crucial, this analysis explores whether the resolution and detail of the 3D models used in point-based prediction impact tracking-based iGNSS QoS prediction.

Figures 5.11 and 5.15 report the errors between the prediction and GPS baselines for driving and walking, respectively, with different granularities of TINs. The results show that generally the 1-zone LOS calculation produced the least RMSE while the 3-zone LOS calculation produced the largest RMSE. Using 1-zone LOS calculation with the bare-earth surface seems acceptable for open sky driving routes but it was incapable of predicting QoS drops in obstructed areas. As the QoS of the GPS baselines is high, close to 1.0 even operated under difficult environments (as previously discussed), we observed minimal errors when using 1-zone LOS calculation and higher errors when using 2-zone and 3-zone LOS calculations. Using 3-zone LOS calculation provided the worst prediction results comparing to GPS baselines. One reason is that even though low iGNSS QoS caused by minor blockage were theoretically predicted, in practice, the GPS receivers were able to minimize noises and improve positional accuracy by employing filters.

After examining differences among the prediction results, it was revealed that the values of the four tracking-based QoS parameters increased when using the 3-zone, 2-zone, and 1-zone

LOS calculations, respectively. Using only the bare-earth TIN in the 1-zone LOS calculation for LOS calculation always produces the best QoS value (1.0) for all parameters as shown in Figures 5.10 and 5.14. On the other hand, using the 2-zone and 3-zone LOS calculations were able to estimate drops in iGNSS QoS especially for H and M driving routes, as shown in Table 5.6 (2-zone) and Table 5.5 (3-zone). Comparing between the prediction results when using the 2-zone and 3-zone LOS calculations, the Friedman tests revealed significant differences on $\widetilde{Ac}$, *Re*, and *Co* for O and H driving routes and walking routes ($p < 0.05$). The 2-zone LOS calculation allows the prediction to focus on major QoS drops due to high elevation objects (> 20 m) but to discard minor drops due to low elevation objects ($\leq$ 20 m). One advantage of the 2-zone LOS calculation is the reduced amount of 3D data that are needed to be retrieved for LOS calculations, especially for open sky areas, which result in less computation time compared to the use of 3-zone LOS calculation.

Table 5.6. Statistics of tracking-based iGNSS QoS prediction (using 2-zone LOS calculation), GPS QoS, and RMSE

			Prediction (Median)				GPS[2] (Median)				RMSE			
Travel Mode	Type	N[1]	$\widetilde{Av}$	$\widetilde{Ac}$	*Re*	*Co*	$\widetilde{Av}$	$\widetilde{Ac}$	*Re*	*Co*	$\widetilde{Av}$	$\widetilde{Ac}$	*Re*	*Co*
Driving	O	72	1.00	0.96	1.00	1.00	1.00	0.95	1.00	1.00	0.07	0.12	0.12	0.15
(sampling = 60	H	45	0.59	0.44	0.27	0.23	1.00	0.83	0.98	0.96	0.52	0.42	0.66	0.62
m, σ = 0.4)	M	45	1.00	0.78	0.90	0.69	1.00	0.90	1.00	1.00	0.13	0.25	0.32	0.33
Walking (sampling = 30 m, σ = 0.6)	M	369	1.00	0.86	1.00	1.00	1.00	0.91	1.00	1.00	0.18	0.27	0.40	0.42

[1] Number of route segments; [2] Average of R1 and R2

Based on the experimental results, using 2-zone LOS calculation is generally sufficient for tracking-based iGNSS QoS prediction for driving routes. The 3-zone LOS calculation, however, seems more appropriate for walking routes to predict changes of *Re* and *Co*. This is because surrounding objects along sidewalks are closer to the walker and the walking speed is much lower than the driving speed. The received GNSS signals in walking mode are potentially

occluded and attenuated more often than that of driving mode. It is expected that iGNSS QoS on walking routes to be lower than driving route under the same surrounding environments.

Minimum Required Accuracy ($\sigma_{threshold}$)

The minimum accuracy required by navigation applications is the key factor for calculating *Co* and *Re*. Prediction results with different $\sigma_{\text{threshold}}$ are shown in Figures 5.9 and 5.13. *Re* is generally greater than *Co*. *Co* is equal to *Re* when iGNSS QoS can be maintained on the segment continuously with no disruption.

The statistics of the prediction results and GPS baselines for *Co* and *Re* are summarized in Tables 5.7 and 5.8, respectively. The RMSEs gradually decreased as the accuracy threshold increased, see Figures 5.15(c-j) for walking routes; the similar trend can be observed for driving routes. Overall, the RMSEs are rather high for high-rise building routes due to large differences between the predicted accuracy range and the accuracy range of the baselines. Although $\sigma_{\text{threshold}}$ = 0.2 produced the minimum errors, it is not appropriate especially for walking routes where the range of the predicted *Co* and *Re* are generally too high, as shown in Figures 5.9 and 5.13, to differentiate good and poor continuity and reliability. Based on the empirical results, $\sigma_{\text{threshold}}$ = 0.4 seems appropriate for driving routes and $\sigma_{\text{threshold}}$ = 0.6 for walking routes for differentiating *Co* and *Re* between open sky and obstructed segments.

Table 5.7. Statistics of *Co* (using 3-zone LOS calculation)

		Co											
		Prediction (Median)				GPS[1] (Median)				RMSE			
Travel Mode	Type	σ= 0.8	σ= 0.6[b]	σ= 0.4[a]	σ= 0.2	σ= 0.8	σ= 0.6	σ= 0.4	σ= 0.2	σ= 0.8	σ= 0.6	σ= 0.4	σ= 0.2
Driving (sampling = 60 m)	O	0.36	0.74	1.00	1.00	1.00	1.00	1.00	1.00	0.59	0.34	0.25	0.15
	H	0.00	0.00	0.04	0.23	0.55	0.83	0.96	1.00	0.53	0.62	0.69	0.66
	M	0.09	0.34	0.52	0.75	0.75	1.00	1.00	1.00	0.50	0.39	0.34	0.28
Walking (sampling = 30 m)	M	0.00	0.74	1.00	1.00	1.00	1.00	1.00	1.00	0.63	0.49	0.43	0.35

[1] Average of R1 and R2; [a] The recommended threshold for driving route; [b] The recommended threshold for walking route

Table 5.8. Statistics of *Re* (using 3-zone LOS calculation)

		Re											
		Prediction (Median)				GPS[1] (Median)				RMSE			
Travel Mode	Type	σ= 0.8	σ= 0.6[b]	σ= 0.4[a]	σ= 0.2	σ= 0.8	σ= 0.6	σ= 0.4	σ= 0.2	σ= 0.8	σ= 0.6	σ= 0.4	σ= 0.2
Driving (sampling = 60 m)	O	0.54	0.90	1.00	1.00	1.00	1.00	1.00	1.00	0.53	0.26	0.18	0.11
	H	0.00	0.00	0.04	0.24	0.70	0.94	0.98	1.00	0.57	0.68	0.73	0.68
	M	0.16	0.75	0.86	0.92	0.93	1.00	1.00	1.00	0.45	0.33	0.32	0.25
Walking (sampling = 30 m)	M	0.00	0.62	1.00	1.00	1.00	1.00	1.00	1.00	0.64	0.48	0.43	0.35

[1] Average of R1 and R2; [a] The recommended threshold for driving route; [b] The recommended threshold for walking route

Different Time of Travel

As GNSS satellites positions change dynamically with time, the impact of time change on the tracking-based iGNSS QoS parameters was analyzed. The sample prediction results comparing to the GPS baselines are shown in Figure 5.10 for driving and in Figure 5.14 for walking.

For driving mode, the Friedman tests showed no significant differences on the four parameters among the prediction results and among the GPS baselines with change of travel time (both $p > 0.05$). There are several reasons for this. First, the length of the driving route segments is rather long comparing to the walking route segments, generally greater than 300 m (see Table 5.4). The tracking-based parameters of the long segments were smoothed out by the weighted average functions (for $\widetilde{Av}$ and $\widetilde{Ac}$) and the ratio functions (for *Co* and *Re*) more than that of the short segments. Second, fluctuations of iGNSS QoS predicted on a street is less than those predicted on the side of the street (i.e., on sidewalks). This is because the distance to surrounding objects (e.g., buildings or trees) from the vehicle on streets is farther than that of the walker on sidewalks. This makes the occluding elevation of the satellites for driving mode lower than for walking mode, which increases the chances for driving mode to maintain good iGNSS QoS.

For walking mode, the Friedman tests revealed significant differences on the four parameters among the prediction results with change of travel time in the prediction ($p < 0.05$).

One reason for this is that surrounding objects along sidewalks play a major role on masking satellites.

It is worth noting that this experiment was tested only with the GPS constellation, which has the ground track repeat period of about a day. Using multi constellations having different repeat periods may cause differences on a set of available satellites to the user at any particular time. As a result, the predicted tracking-based parameters, especially for driving routes, may be affected as time of travel changes.

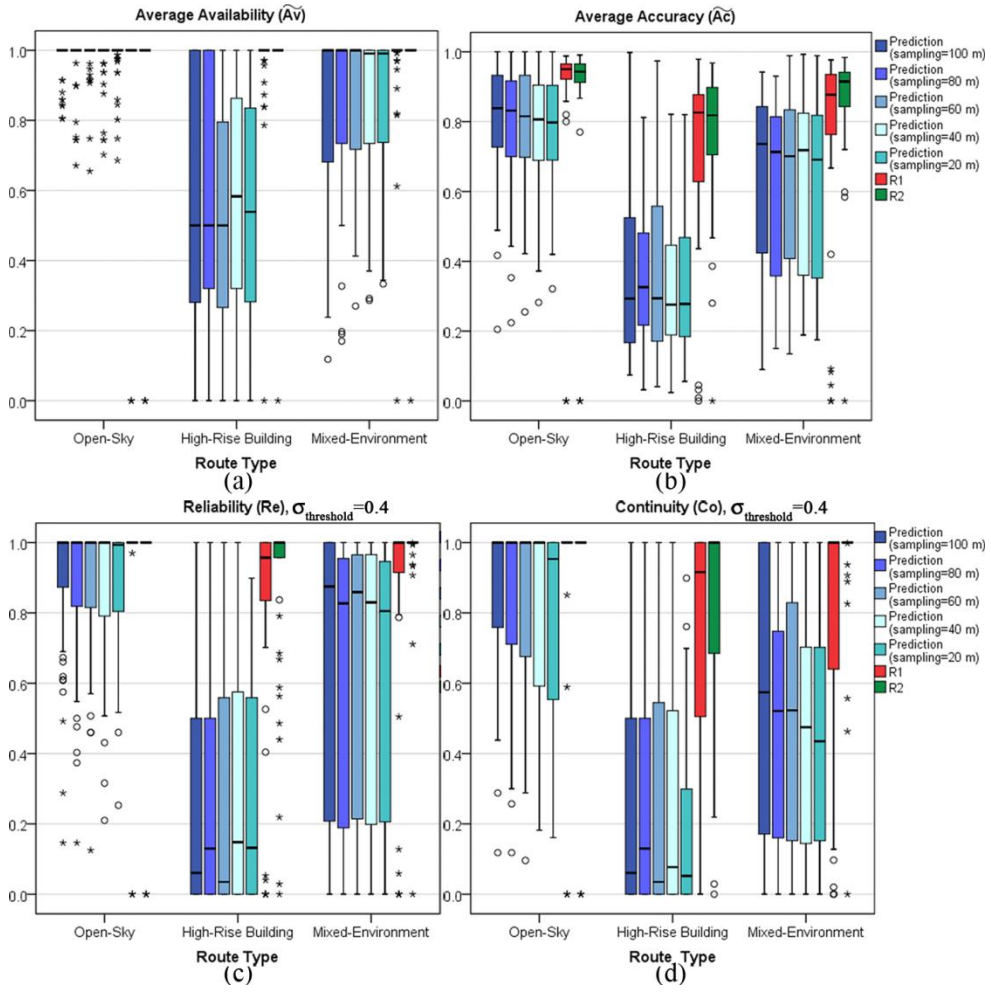


Figure 5.8. Prediction results and GPS QoS baselines for driving routes with different sampling distances (using 3-zone LOS calculation)

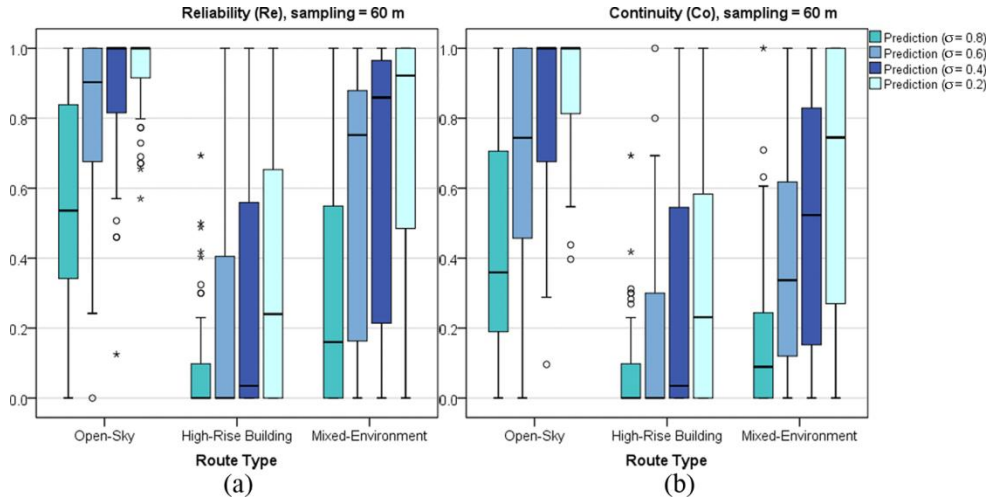


Figure 5.9. Reliability and continuity results for driving routes with different minimum accuracy requirements (fixed sampling distance = 60 m and 3-zone LOS calculation)

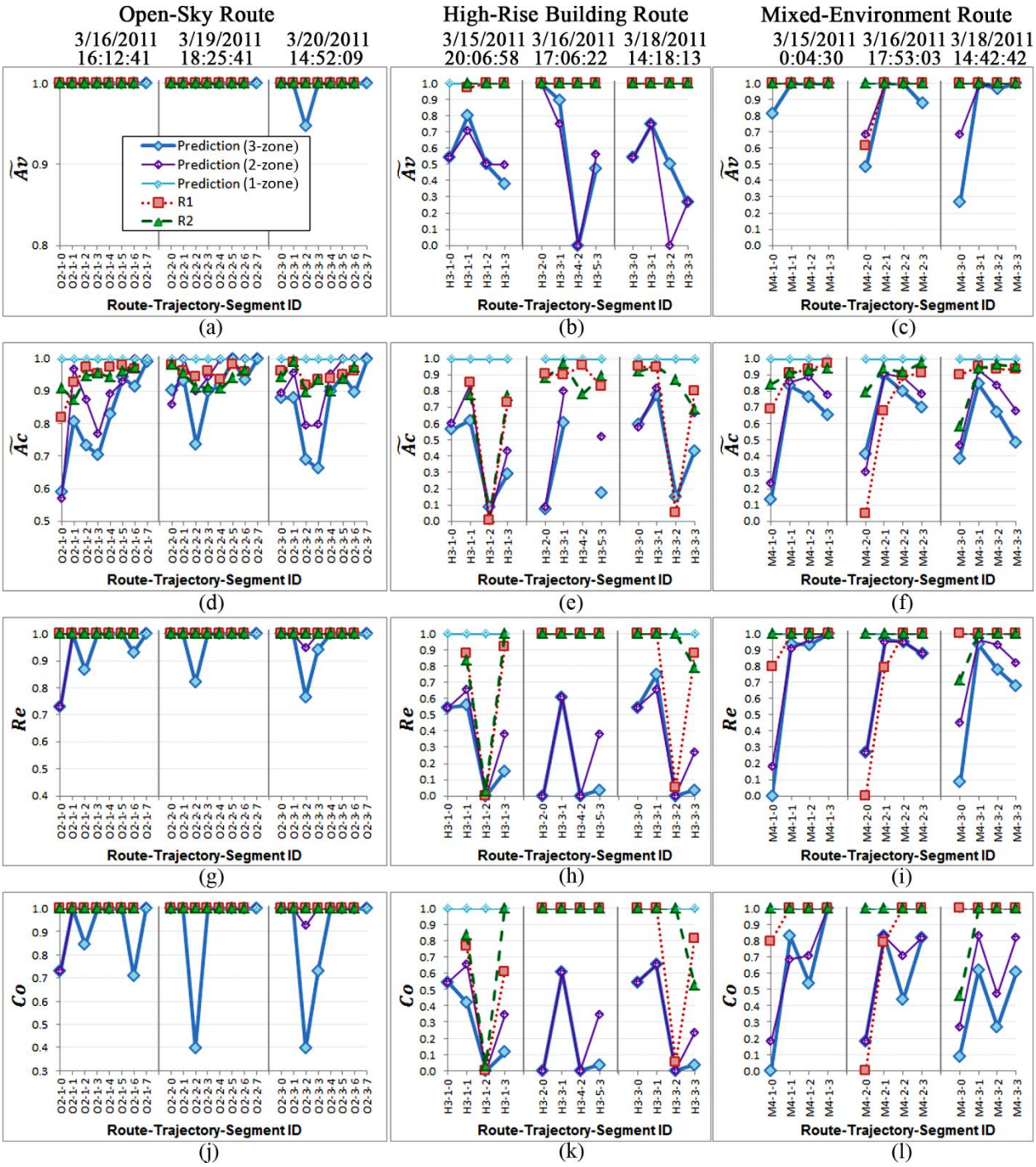


Figure 5.10. iGNSS QoS results and GPS QoS baselines for driving routes at three different times of travel (fixed sampling distance = 60 m, and $\sigma_{threshold} = 0.4$)

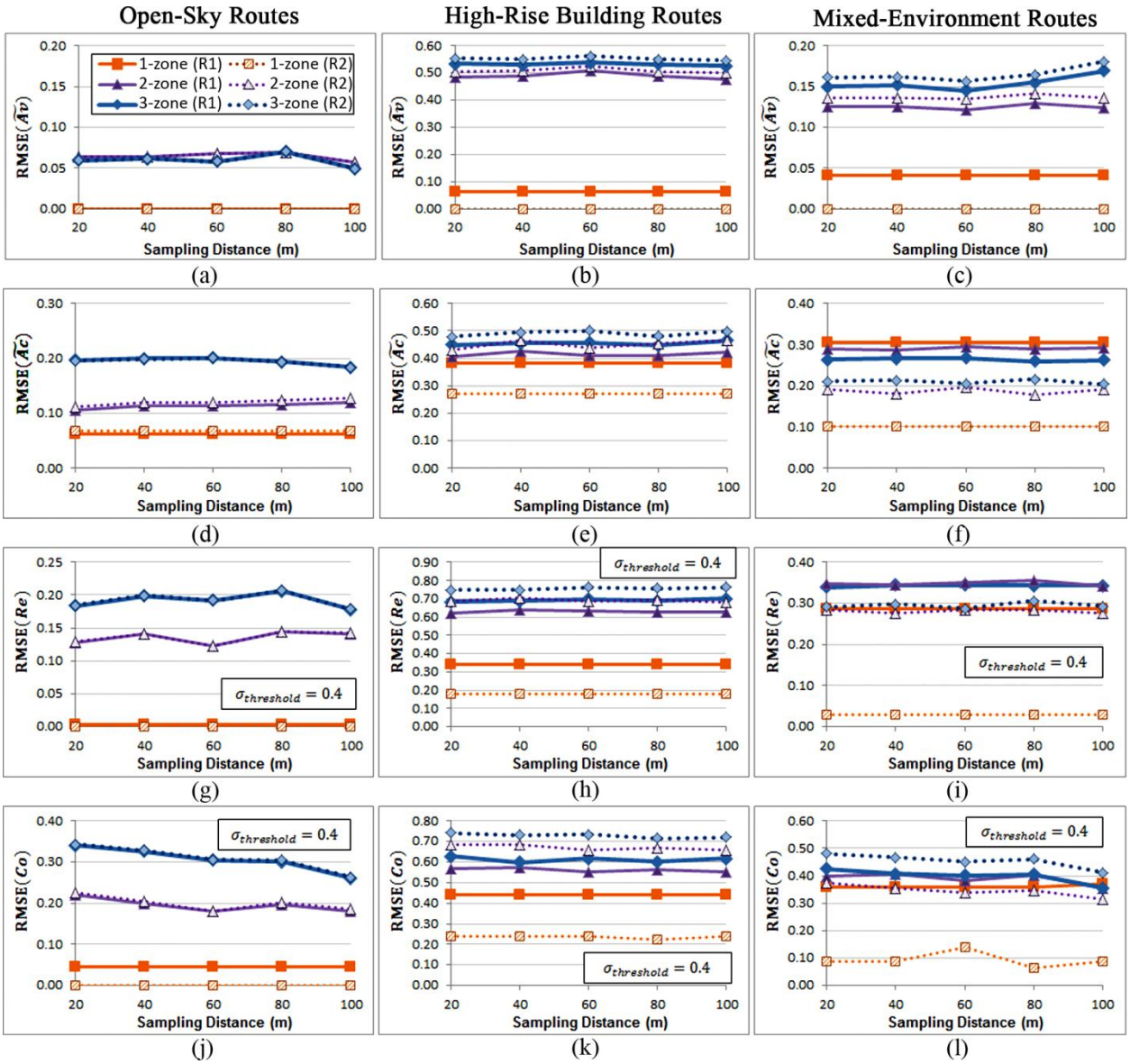


Figure 5.11. RMSEs of the navigation-based iGNSS QoS parameters for driving routes

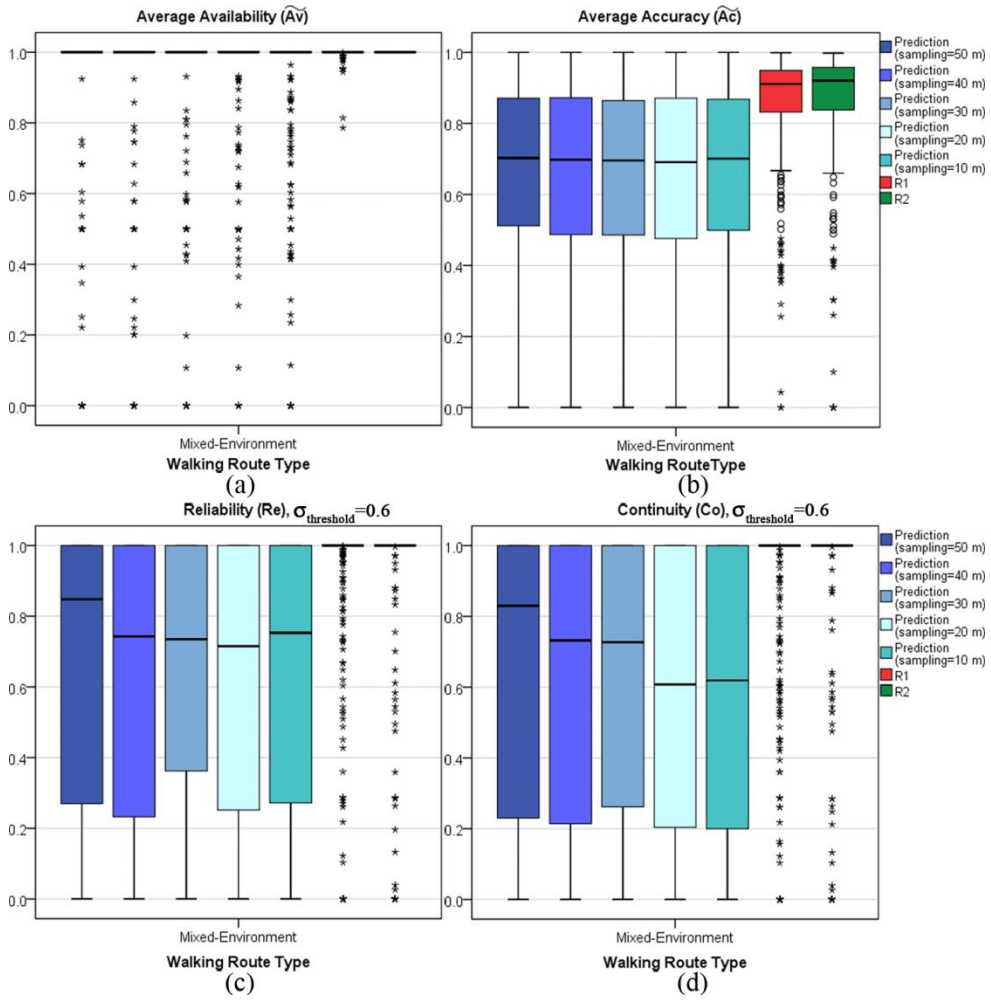


Figure 5.12. Prediction results and GPS QoS baselines for the walking routes with different sampling distances (using the 3-zone)

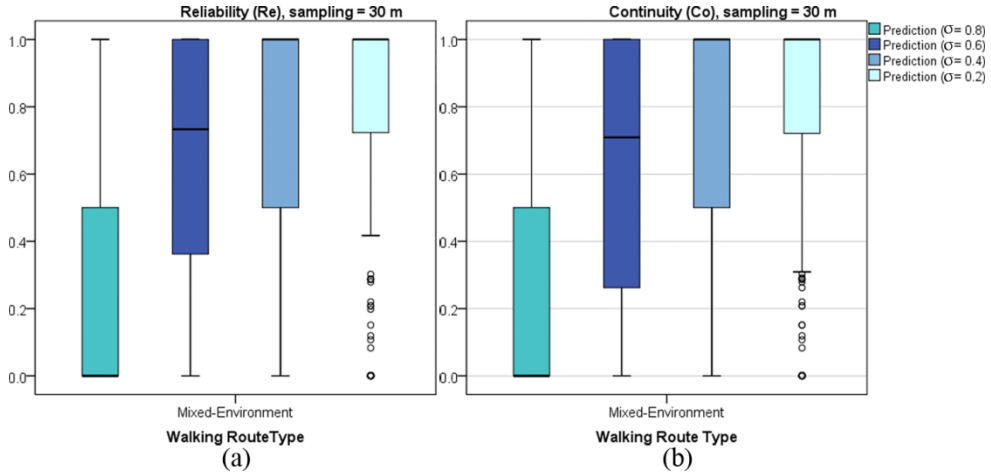


Figure 5.13. Reliability and continuity results for walking routes with different minimum accuracy requirements (fixed sampling distance = 30 m and 3-zone LOS calculation).

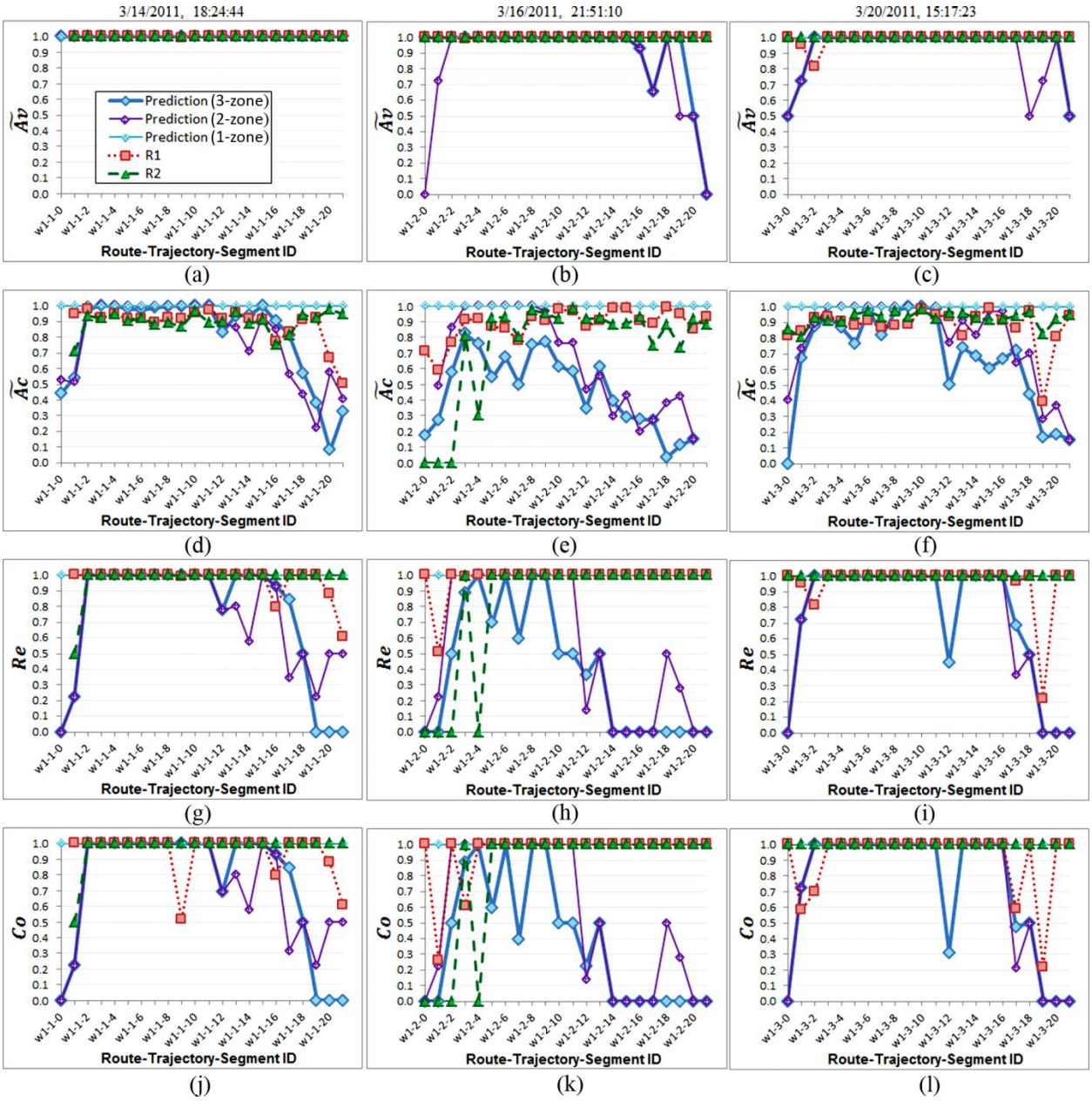


Figure 5.14. Example of iGNSS QoS results and GPS QoS baselines for walking routes at three different times of travel (fixed sampling distance = 30 m, and $\sigma_{threshold} = 0.6$)

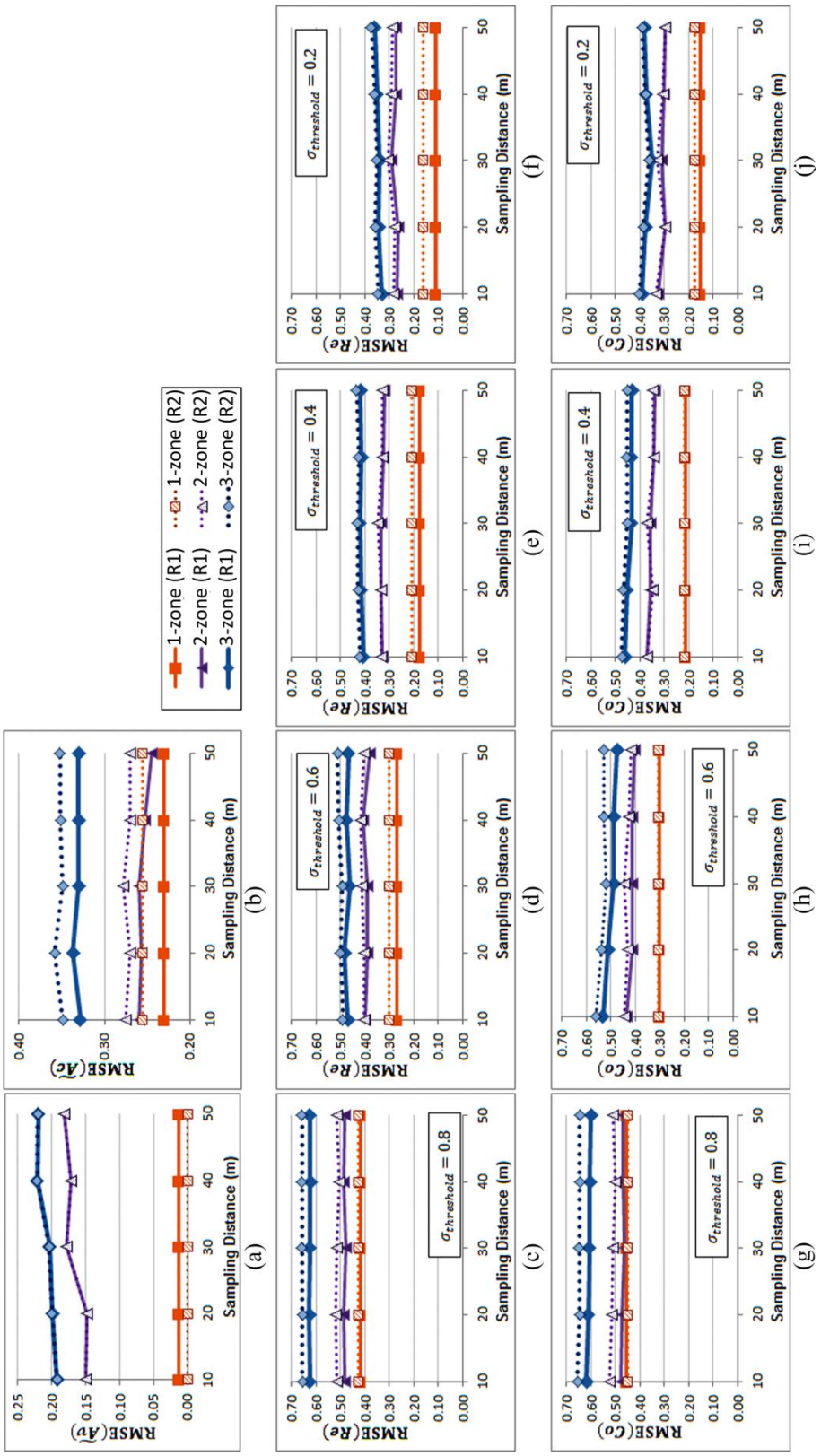


Figure 5.15. RMSEs of the navigation-based iGNSS QoS parameters for walking routes

5.5.4.2 Analysis II: iGNSS QoS Segmentation

The prediction results and GPS baseline for iGNSS QoS segmentation are shown in Figures 5.16 – 5.19 for driving routes and in Figures 5.20 – 5.23 for walking routes.

Sampling Distance

Overall, varying sampling distance caused minor differences on the errors of $\widetilde{Ac}$ on chunks, see Figures 5.17(a-c) and Figure 5.21(a) for RMSE of the prediction results when τ = 0.4. However, it was found out that the results of $\widetilde{Ac}$ on chunks were significantly different from the GPS baselines. A set of histograms showing the results of $\widetilde{Ac}$ on chunks is shown in Figure 5.16 for driving and Figure 5.20 for walking. Typically, the prediction results were relatively lower than the GPS baselines, especially in obstructed areas. This caused high RMSE for high-rise building routes and walking routes but low RMSE the open sky routes, as statistically summarized in Table 5.9 for a fixed sampling distance and τ. In addition, the prediction results provided a high number of chunks with unavailable position solutions (N/A). This was another factor that causes large errors in those routes passing by obstructed areas.

The Kruskal-Wallis tests showed no significant difference on the predicted $\widetilde{Ac}$ on chunks for open sky and highly obstructed routes (H(4) = 2.57 and 6.94, $p > 0.05$) but revealed significant differences for mixed environment routes and walking routes (H(4) = 11.74 and 9.75, $p < 0.05$) upon change of sampling distances. Significant differences occurred due to the results of the shortest and longest sampling distance used in the experiment. Generally, increasing the sampling distance tended to slightly increase $\widetilde{Ac}$ on chunks, as shown in Figures 5.17(d-f) for driving and Figure 5.21(b) for walking.

However, the Kruskal-Wallis tests showed that sampling distance significantly influences length of chunks for all route types ($p < 0.05$), except open sky routes. Increasing the sampling distance tended to increase the length of chunks while decreasing the number of chunks, as shown in Figures 5.17(g-l) for driving and Figures 5.21(c-d) for walking. Table 5.9 summarizes statistics of the results. With these empirical results, it is preferable to apply a long sampling distance for open sky routes (e.g., 200 to 300 m). Under mixed environment or obstructed areas, 60-m and 30-m sampling distances seem acceptable for driving and walking routes, respectively.

Table 5.9. Statistics of iGNSS QoS on chunks (using 3-zone LOS calculation)

Mode	Type	N[1]	Length (Median) [m]	% Chunks with N/A		$\widetilde{Ac}$ (Median)		RMSE $\widetilde{Ac}$
				Prediction	GPS[2]	Prediction	GPS[2]	
Driving (sampling = 60 m, τ = 0.4)	O	87	311.40	10.34%	4.02%	0.81	0.94	0.23
	H	146	60.00	45.21%	9.25%	0.27	0.84	0.62
	M	113	120.00	34.51%	3.54%	0.53	0.87	0.30
Walking (sampling = 30 m, τ = 0.4)	M	439	18.60	12.98%	4.56%	0.69	0.91	0.37

[1] Number of chunks; [2]Average of R1 and R2

Variation Tolerance (τ)

Variation tolerance assigned to SMA has a slight impact on RMSE of $\widetilde{Ac}$ on chunks, as shown in Figures 5.18(a-c) for driving and Figure 5.22 (a) for walking. The errors remained low for open sky routes and high for obstructed routes. A slight decrease in RMSE can be observed as the tolerance threshold increases.

Figures 5.18(d-f) and 5.22(b) show the medians of $\widetilde{Ac}$ on chunks with different tolerance values. A Kruskal-Wallis test showed no significant differences on $\widetilde{Ac}$ on chunks for open sky routes ($H(4) = 3.62$, $p > 0.05$). It reflected that iGNSS QoS has low variations in open sky areas. For mixed environment driving routes, the Kruskal-Wallis test revealed a significant change on $\widetilde{Ac}$ on chunks ($H(4) = 66.57$, $p < 0.05$), particularly when $\tau = 0.1$.

Setting a small variation tolerance tended to increase the length of chunks and the total number of chunks as shown in Figures 5.18(g-l) and 5.22(c-d). The Kruskal-Wallis tests showed significant differences on the chunk length for all route types ($p < 0.05$), except for high-rise building routes ($H(4) = 3.38$, $p > 0.05$). Changing the τ value from 0.1 to 0.4 reduced the total number of chunks more than half for open sky and mixed environment routes and by half for walking routes. The average chunk lengths remained unchanged when τ was 0.4 and 0.5. This in turn revealed that sampling distance can be as large as 311.4 m for open sky routes, 120 m for mixed environment routes, and 18.6 m for walking routes. For high-rise building routes, on the other hand, no conclusion could be drawn; it is highly dependent on sampling distance.

TIN Data Layers

Similar to the findings in Section 5.5.4.1, prediction using 1-zone LOS calculation provides the least RMSE. Again, this was due to optimistic positions estimated by the GPS receivers. The RMSE of the 2-zone LOS calculation is slightly lower than that of the 3-zone LOS calculation, as shown in Figures 5.17 – 5.18 for driving and Figures 5.21 – 5.22 for walking. Table 5.10 statistically summarizes the results of the 2-zone LOS calculation, compared to the results of the 3-zone LOS calculation in Table 5.9.

Comparing among the predicted $\widetilde{Ac}$ on chunks when using 2-zone and 3-zone LOS calculations, the Kruskal-Wallis tests revealed that the results of the 2-zone LOS calculation were significantly higher for open sky routes ($H(4) = 13.86$, $p < 0.05$) and walking routes ($H(4) = 36.26$, $p < 0.05$). Thus, it is generally acceptable to use 2-zone LOS calculation for open sky routes since vehicles travel at a high speed where the GPS filter has a high confidence on its position solutions. Using 2-zone LOS calculation causes the prediction to ignore minor obstacles in moderately obstructed areas. On the other hand, walking speed is usually low (< 3 m/s), where large position errors can be observed, especially in obstructed areas (Ochieng et al., 2003). Using 3-zone LOS calculation is more appropriate to predict drops in iGNSS QoS on walking routes in mixed environment areas, even though the experimental results showed higher RMSE than 2-zone LOS calculation. Large errors mostly occurred due to the difference between the prediction value range and GPS QoS value range in obstructed areas.

Granularity of TIN data layers has slight impact on the average length of chunks and the number of chunks per route for all route types ($p > 0.05$). The results are shown in Figures 5.17 – 5.18 for driving and Figures 5.21 – 5.22 for walking. The chunk length when using 1-zone LOS calculation is equal to the size of the route segment as $\widetilde{Ac}$ is always predicted as 1.0. With variation of iGNSS QoS predicted by using the 2-zone and 3-zone LOS calculations, the average chunk length is usually shorter than the route segments.

Table 5.10. Statistics of iGNSS QoS on chunks (using 2-zone LOS calculation)

		N^1	Length (Median)	% Chunks with N/A		Ac (Median)		RMSE
Mode	Type		[m]	Prediction	GPS[2]	Prediction	GPS[2]	Ac
Driving	O	88	297.60	10.23%	4.55%	0.96	0.95	0.18
(sampling = 60	H	138	63.50	46.38%	7.61%	0.28	0.83	0.58
m, t = 0.4)	M	120	120.00	35.00%	3.33%	0.46	0.88	0.30
Walking (sampling = 30 m, t = 0.4)	M	448	18.90	11.83%	4.80%	0.86	0.91	0.31

[1] Number of chunks; [2]Average of R1 and R2

Different Time of Travel

Even though the tracking-based parameters on route segments may not strongly indicate changes of iGNSS QoS as the travel time changes, as discussed in Section 5.5.4.1, iGNSS QoS chunks can effectively inform changes of iGNSS QoS spatially and temporally, especially in problematic areas along the route. Examples of iGNSS QoS chunks for a driving route and a walking route are shown in Figures 5.19 and 5.23, respectively. For visualization on a map, $\widetilde{Ac}$ on chunks were classified into four groups: Good ($0.8 < \widetilde{Ac} \leq 1.0$), Fair ($0.4 < \widetilde{Ac} \leq 0.8$), Poor ($0 \leq \widetilde{Ac} \leq 0.4$), and N/A. Generally, both GPS receivers provided high accurate positions on Good QoS chunks. Large deviations of GPS points mostly occurred on Poor and N/A QoS chunks. Changes on iGNSS QoS usually occurred in moderately blocked areas.

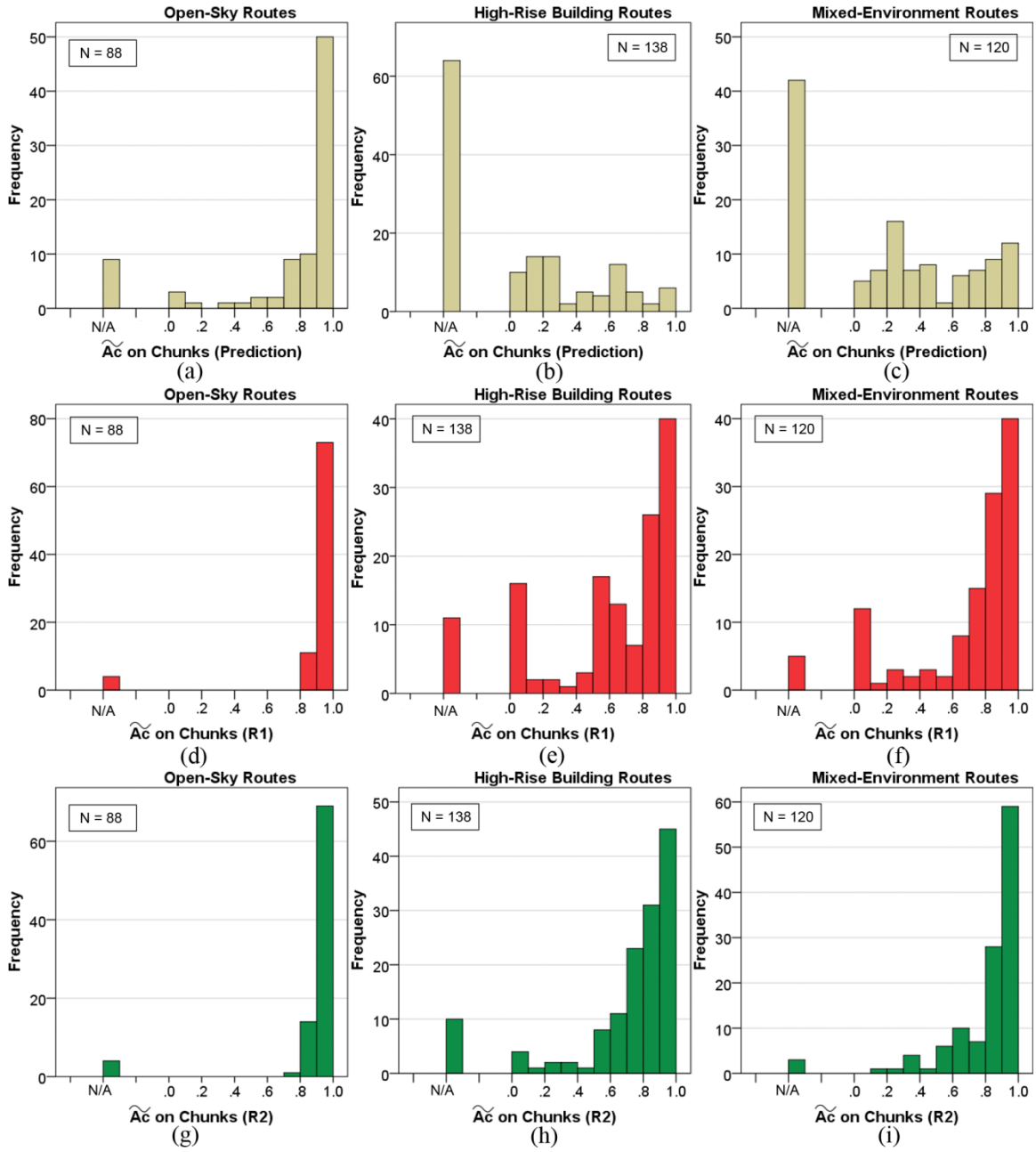


Figure 5.16. Statistics of $\widetilde{Ac}$ on chunks for driving routes (fixed sampling = 60 m, τ = 0.4, and 2-zone LOS calculation for prediction)

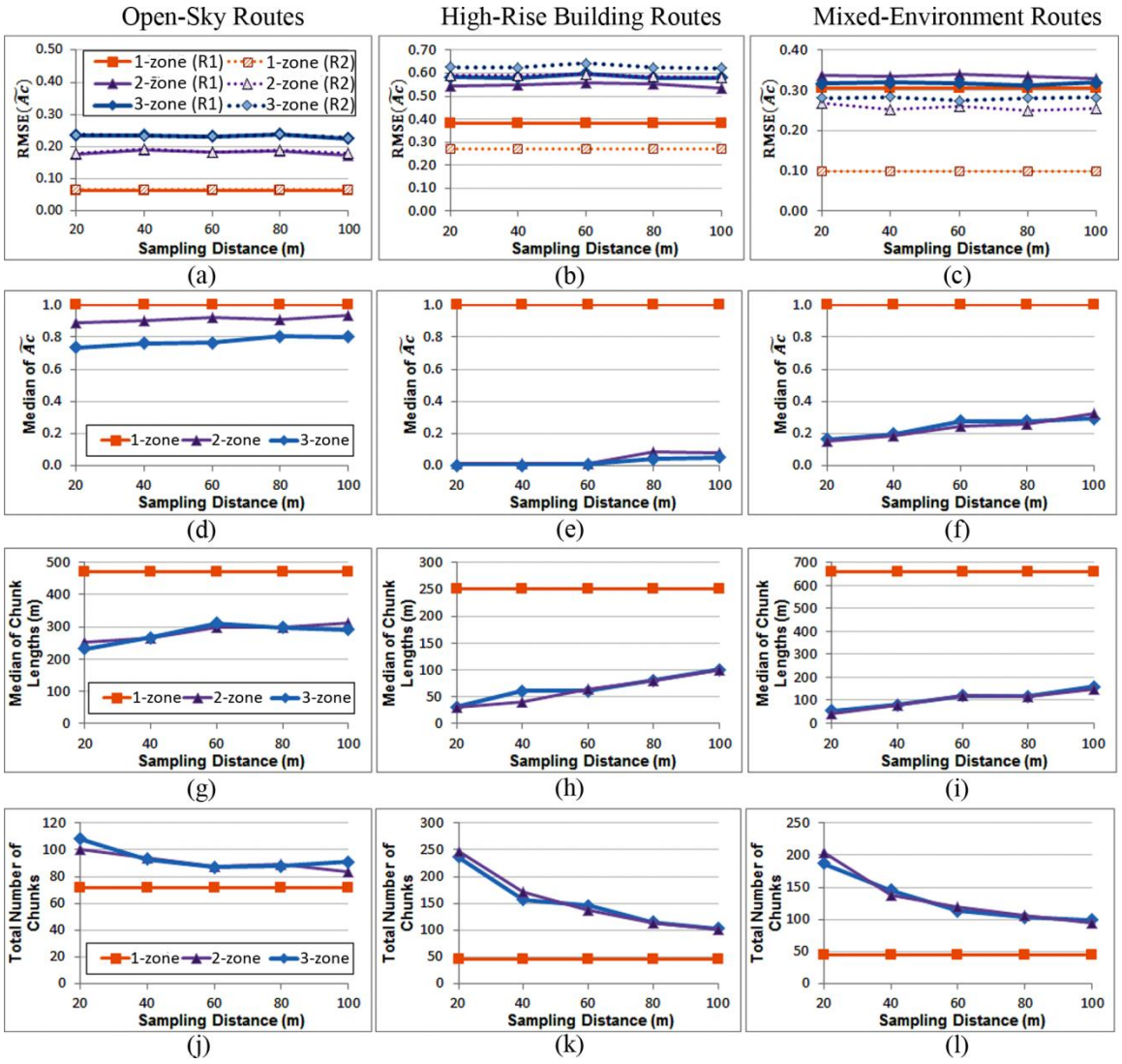


Figure 5.17. Statistics of $\widetilde{Ac}$ on chunks and chunks for driving routes (fixed $\tau = 0.4$)

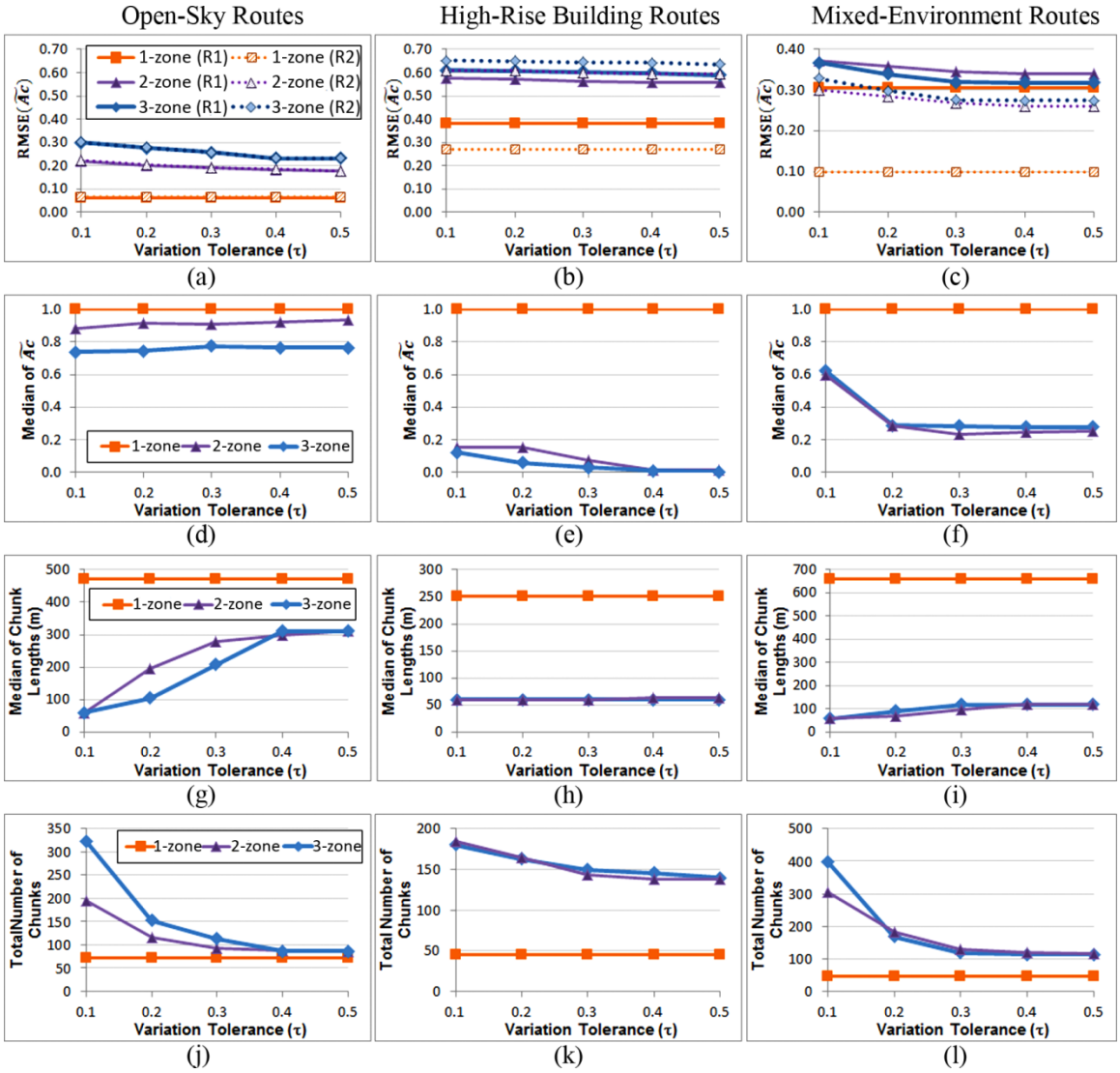
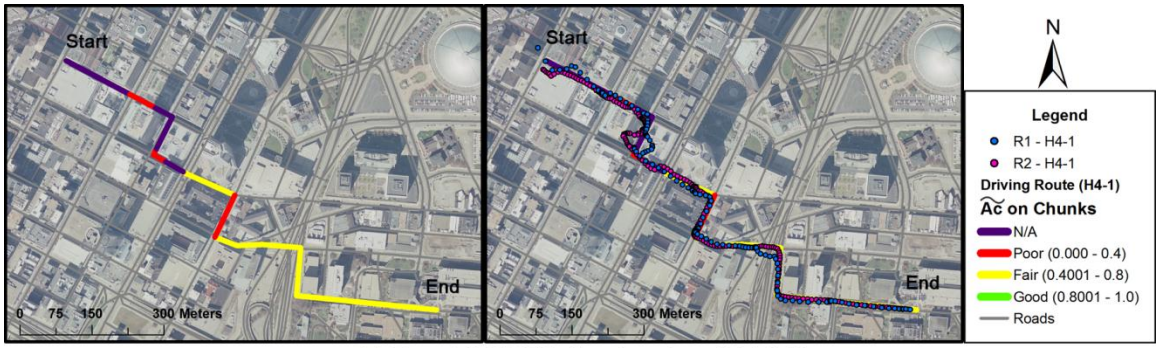
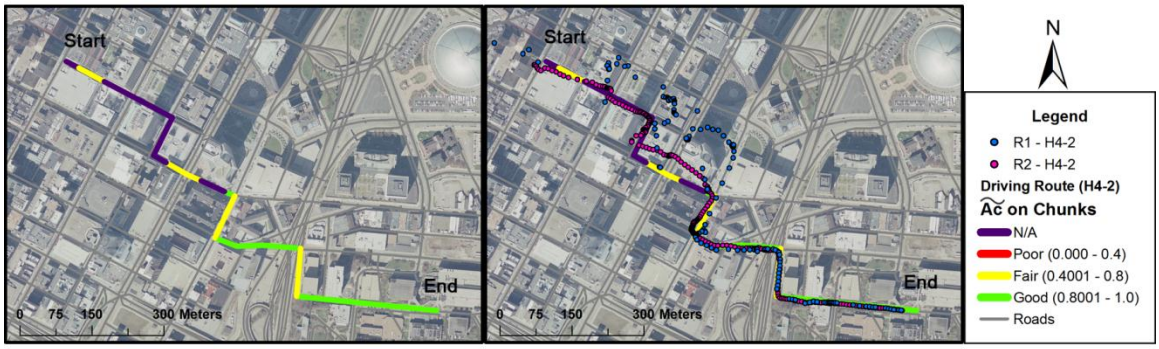


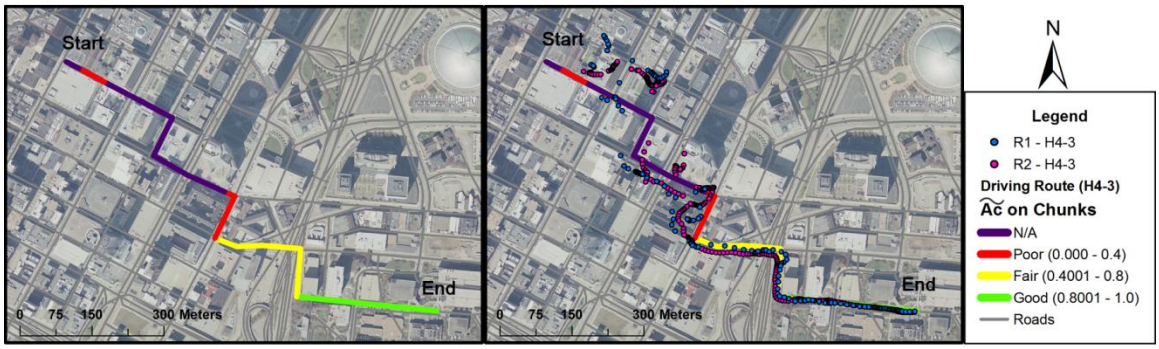
Figure 5.18. Statistics of $\widetilde{Ac}$ on chunks and number of chunks for driving routes (fixed sampling distance = 60 m)



(a) Date 3/16/2011, Starting time 17:15:09



(b) Date 3/18/2011, Starting time 14:25:05



(c) Date 3/19/2011, Starting time 15:53:58

Figure 5.19. Example of iGNSS QoS chunks and GPS points on a driving route at three different times of travel (fixed sampling distance = 60 m, $\tau = 0.4$, and 2-zone LOS calculation)

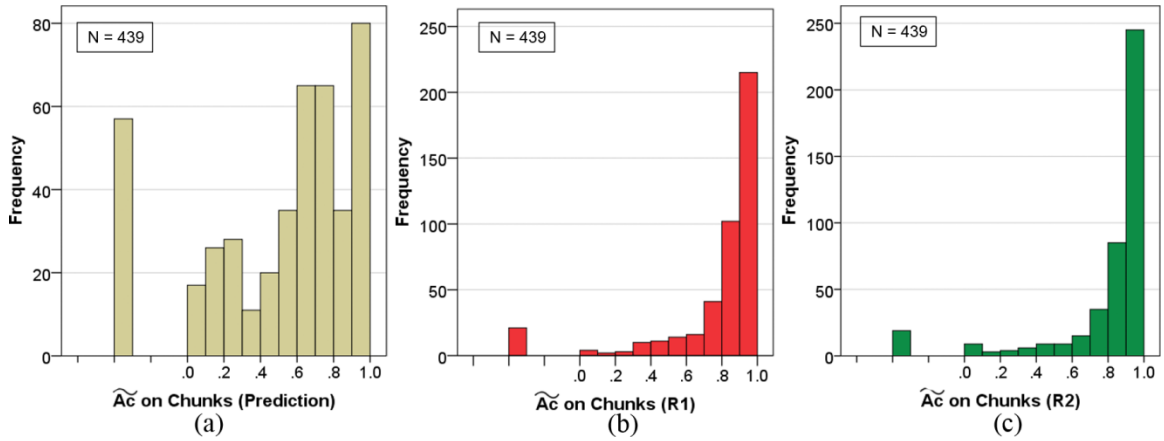


Figure 5.20. Statistics of $\widetilde{Ac}$ on chunks for walking routes (fixed sampling = 30 m, τ = 0.4, and 3-zone LOS calculation for prediction)

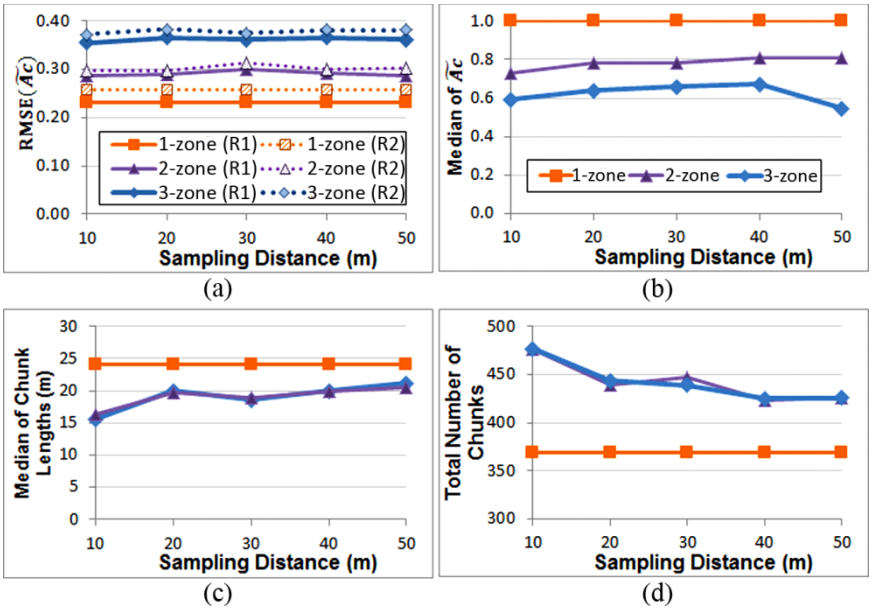


Figure 5.21. Statistics of $\widetilde{Ac}$ on chunks and chunks for walking routes (fixed τ = 0.4)

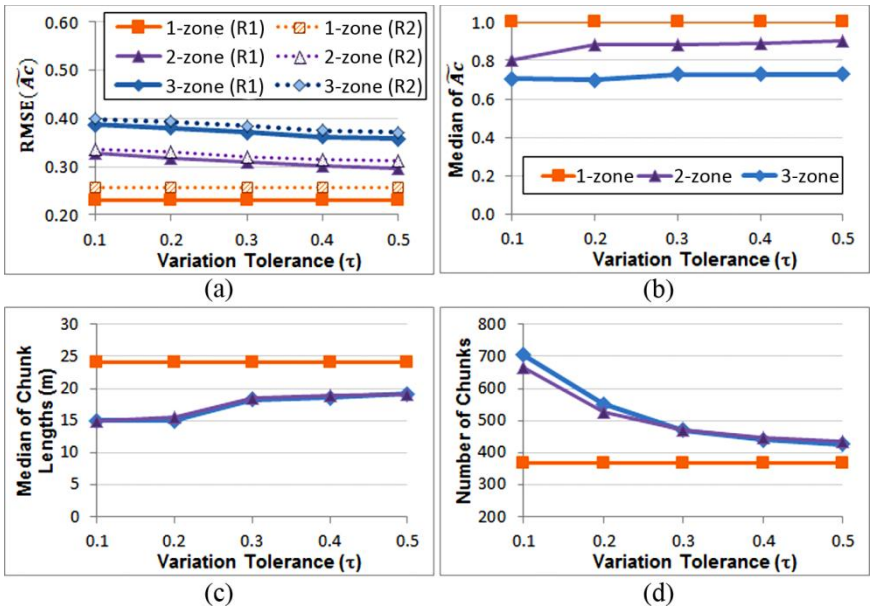
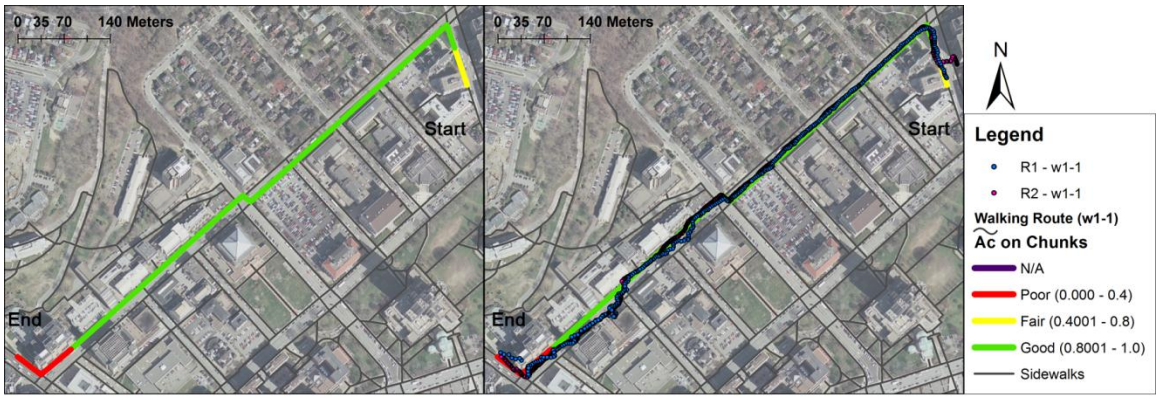
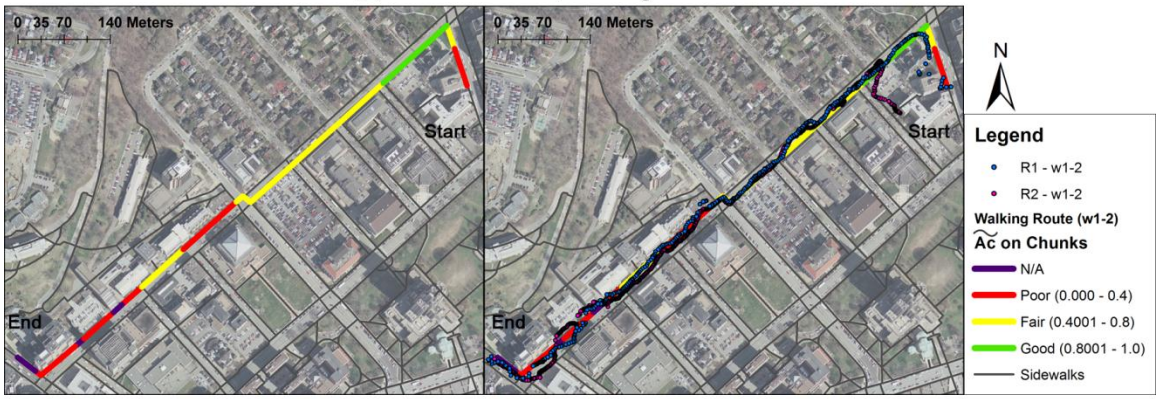


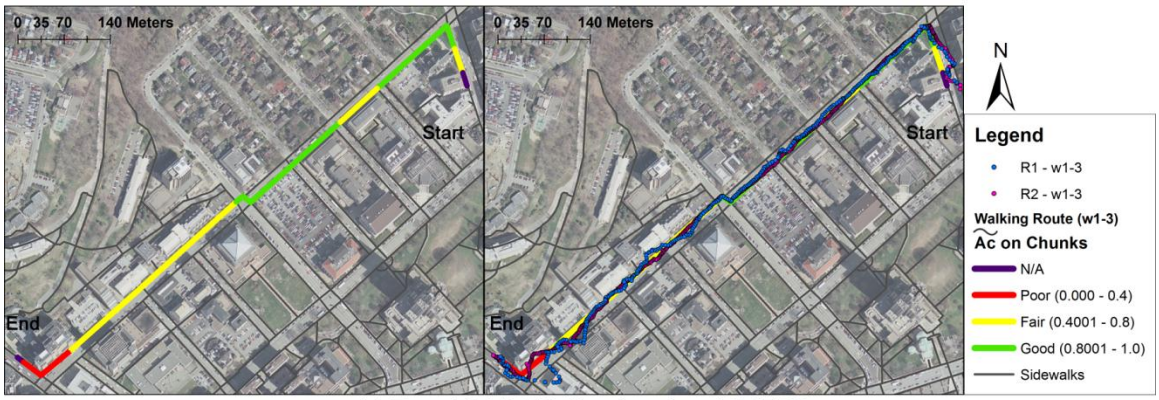
Figure 5.22. Statistics of $\widetilde{Ac}$ on chunks and number of chunks for walking routes (fixed sampling distance = 30 m)



(a) Date 3/14/2011, Starting time 18:24:44



(b) Date 3/16/2011, Starting time 21:51:10



(c) Date 3/20/2011, Starting time 15:17:23

Figure 5.23. Example of iGNSS QoS chunks and GPS points on a walking route at three different times of travel (fixed sampling distance = 30 m, τ = 0.4, and 3-zone LOS calculation)

6.0 EFFICIENT ALGORITHM AND DISTRIBUTED PROCESSING FOR REAL-TIME PREDICTION

6.1 INTRODUCTION

Although the process of examining visibility of GNSS signals in point-based iGNSS QoS prediction is straightforward by using ray tracing techniques, the amount of surface data included in each LOS determination is typically large. This problem exacerbates when high-resolution surface data is used in order to achieve highly accurate prediction. In addition, the availability of iGNSS satellites from different constellations increases the number of LOSs in which its visibility must be determined at a given location and time. In the context of real-time navigation, computing visibility of each satellite and other QoS parameters at every sample point within a short time is even more challenging.

To address data- and compute-intensive tasks of prediction iGNSS QoS for real-time navigation systems/services, two approaches were taken. The first approach is to check visibility for a subset of satellites that can provide a near-optimal solution when all visible satellites are used. This is to minimize number of LOS calculations. An algorithm, Multi-Constellation Satellite Selection Algorithm (MCSSA), was developed to select a subset of satellites from a pool of available iGNSS satellites (Roongpiboonsopit and Karimi, 2009). MCSSA shows that using eight satellites selected by this algorithm can provide a near-optimal solution with low computation time. Thus, point-based iGNSS QoS prediction can prioritize the LOS determination tasks by first checking visibility of the selected satellites. If all selected satellites are visible, the simulation can shorten the visibility calculations and provide quick response. If not, the algorithm needs to choose additional satellites that provide near-optimal solutions to check visibility.

The second approach is to utilize distributed computing platforms, i.e., grids and clouds. Given the volumes of data, sequential processing on any single computer is not sufficient to carry out the tasks of predicting iGNSS QoS under a short time constraint. Grid-based processing is considered as a viable approach since it supports distributed computing that can significantly reduce overall computing time by making effective use of large-scale, parallel resources. Today, various grid computing platforms (e.g., TeraGrid[7], Open Science Grid[8]) are available to researchers and practitioners to submit data and/or computationally-intensive jobs. In this dissertation, the University of Pittsburgh's campus grid, called PittGrid, is investigated for the purpose of the iGNSS QoS prediction (Karimi et al., 2011c). Cloud computing is an attractive and alternative platform to other high-performance platforms (e.g., grid or parallel computing). Cloud computing is one of the most promising developments towards the vision of utility computing where a large amount of computing resources and storage can be accessed through the Internet, offering applications an unprecedented, on-demand scalability. Although several research studies (Blower, 2010, Cornillon, 2009, Wang et al., 2009) have shown a viable solution of cloud computing paradigm for data- and compute-intensive geoprocessing tasks, the feasibility of current cloud computing platforms remains unexplored for real-time geoprocessing. This chapter explores the performances and limitations of an existing cloud (i.e., Google App Engine) to accommodate iGNSS QoS prediction (Karimi et al., 2011b).

6.2 EFFICIENT ALGORITM FOR SATELLITE SELECTION

For a single GNSS, only 8-12 satellites may be simultaneously observed at a location. However, for iGNSS, the number of satellites may double, triple, or quadruple, depending on GNSS combinations. The increasing number of available satellites requires more calculations on checking satellite visibility at each sample point. The goal is to reduce visibility computation by considering only a subset of the satellites that will provide a near-optimal solution.

MCSSA uses the geometry of satellites as a criterion and was developed based on two existing satellite selection algorithms: the Maximized Volume Algorithm (MVA) (Kihara and

[7] http://www.teragrid.org
[8] http://www.opensciencegrid.org

Okada, 1984) and the Quasi-Optimal satellite selection Algorithm (QOA) (Park, 2001). The advantages of MCSSA are that it has no limitation on number of satellites, provides near-optimal solutions, and requires minimal computing time.

6.2.1 Satellite Selection Algorithms

This section summarizes existing satellite selection algorithms and discusses their strengths and weaknesses. A brief overview of these algorithms along with their strengths and weaknesses is shown in Table 6.1.

Table 6.1. Summary of the satellite selection algorithms

Algorithm Name	Method	Strengths	Weaknesses
Optimal satellite selection algorithm (**OSSA**)	Finds the set of satellites with the minimum DOP	• Ensures the optimal solution • Not limited to the number of selected satellites	• Large problem space ${}_NC_n$ • Intensive computation load
Highest elevation satellite selection algorithm (**HEA**)	Selects the satellites at the highest elevation angles	• Requires a little computation • Not limited to the number of selected satellites	• Poor geometry
Maximum volume algorithm (**MVA**; Kihara and Okada, 1984)	Selects four satellites that maximize the volume of tetrahedron	• Provides a near-optimal solution with reasonable computation load	• Limits only four satellites because of the tetrahedron model
Four-step satellite selection algorithm (Li et al., 1999)	Selects four satellites that maintain the regular tetrahedron shape and minimize the DOP	• Provides a near-optimal solution with reasonable computation load	• Limits only four satellites because of the tetrahedron model
Quasi-optimal satellite selection algorithm (**QOA**; Park, 2001)	Eliminates the redundant satellites until satisfying the predefined number of satellites	• Not limited to the number of selected satellites • Less computation • Provides a near-optimal solution	• Designed for LEO applications

Algorithm Name	Method	Strengths	Weaknesses
Dartboard DOP algorithm (D'Angelo et al., 2005)	Selects twelve satellites falling near the center of twelve zones on the skyview dashboard	• Provides good results when select twelve satellites	• Fixes with twelve satellites

6.2.2 Multi-Constellation Satellite Selection Algorithm

MCSSA is not limited to any number of satellites or to any specific constellation. The logic of this algorithm is to select the set of visible satellites that are scattered evenly over the sky while ensuring a near optimal geometric arrangement. With this, MCSSA combines the strengths of two existing algorithms, MVA and QOA, by expanding the capability of MVA with the redundancy technique used in QOA. However, instead of eliminating the most redundant satellite with respect to all other satellites as in QOA, MCSSA selects the least redundant satellite with respect to previously selected satellites. Figure 6.1 presents the flowchart of MCSSA for selecting *n* satellites (*n* = 1, 2, …, *N*-1).

MCSSA assumes that the integrated GNSS receiver has already received constellations' information and can extract all visible satellite positions with respect to receiver's position. MCSSA starts the process by generating the *N* visible LOS matrix (**H**),

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_N \end{bmatrix} \tag{6.1}$$

where $\mathbf{h}_i = [x_i \quad y_i \quad z_i]$ is a unit vector pointing from the estimated receiver's position to the i^{th} visible satellite (i = 1, 2, …, N).

The first four satellites are selected using MVA. The results of MVA will ensure that the four-core geometrical arrangement gives the near-optimal geometry factor, i.e., PDOP. The next step of MCSSA is to prepare parameters for the following tasks of selecting additional satellites. It removes the four vectors of the selected satellites from **H** and adds these vectors to the selected

selected LOS vector matrix **S**. With this, the remaining LOS vector matrix (**H**) contains $N-4$ vectors, whereas the selected LOS vector matrix (**S**) contains four vectors, expressed as follows: follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_{N-4} \end{bmatrix} \text{ and } \mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \\ \mathbf{s}_4 \end{bmatrix}. \tag{6.2}$$

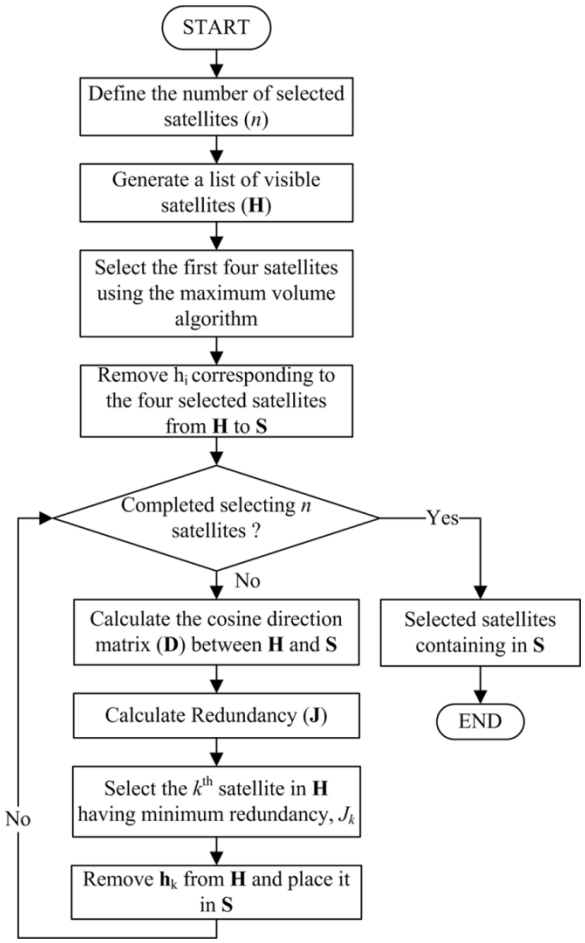


Figure 6.1. Flowchart of MCSSA

In case of selecting more than four satellites, MCSSA sequentially and repeatedly selects an additional satellite until the total of selected satellites reaches the predefined number. The satellite with minimum redundancy value with respect to the previously selected satellites will be added to the selected satellites set. Let m represent the number of satellites that have been selected, so for the first iteration, m = 4. The cost function for determining the redundancy value is a function of the direction cosine between each pair of remaining vectors ($\mathbf{h}_i, i = 1,2, \ldots, N-m$) and all selected vectors ($\mathbf{s}_j, j = 1,2, \ldots, m$). The direction cosine matrix **D** is computed by:

$$\mathbf{D} = \mathbf{H} \times \mathbf{S}^T = \begin{bmatrix} \cos\theta_{11} & \cos\theta_{12} & \cdots & \cos\theta_{1m} \\ \cos\theta_{21} & \cos\theta_{12} & \cdots & \cos\theta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \cos\theta_{(N-m)1} & \cos\theta_{(N-m)2} & \cdots & \cos\theta_{(N-m)m} \end{bmatrix} \tag{6.3}$$

The cost function defined in Equation 6.4 is similar to the cost function used in QOA.

$$J_{ij} = \cos 2\theta_{ij} \tag{6.4}$$

Therefore, the cost approaches 1 when two vectors are close to collinear ($\theta_{ij} \cong 0°$ or $\theta_{ij} \cong 180°$) and -1 when the angle between the two vectors are perpendicular ($\theta_{ij} = \pm 90°$). The redundancy value for each remaining satellite in **H**, J_i , is defined as the sum of the costs between $\mathbf{h}_i$ and all $\mathbf{s}_j$. In other words, the redundancy value is the sum of square of the elements in matrix **D**, expressed as follows:

$$J_i = \sum_{j=1}^{m} \left(\cos 2\theta_{ij}\right) = \sum_{j=1}^{m} \left(2\cos^2\theta_{ij} - 1\right)$$

given $d = \cos\theta_{ij}$

$$J_i = \textstyle\sum_{j=1}^{m} \left(2d_{ij}^2 - 1\right). \tag{6.5}$$

MCSSA selects the k^{th} satellite as the additional satellite, which has the minimum redundancy value by:

$$J_i = \min_i\{J_1, J_2, \ldots, J_{N-m}\} \tag{6.6}$$

Next, MCSSA updates **H** and **S** by removing $\mathbf{h}_k$ from **H** and placing it as $\boldsymbol{s}_{m+1}$ in **S**. So far, the m + 1 satellites have been selected. The process of calculating the redundancy values is repeated until all n satellites are selected ($m + 1 = n$).

However, in order to reduce redundant computations during the calculation of the direction cosine matrix, the new direction cosine matrix (**D**′) can be updated from the old **D** by eliminating the k^{th} row ($\mathbf{D}_k$) and adding the new direction cosine column ($\mathbf{D}_{new}$) that represents the cost between the updated remaining satellites (**H**′) and $\mathbf{s}_{m+1}$.

$$\mathbf{D}_{new} = \mathbf{H}' \cdot \mathbf{s}_{\mathrm{m+1}}^{\mathrm{T}} = \begin{bmatrix} \cos\theta_{1(m+1)} \\ \cos\theta_{2(m+1)} \\ \vdots \\ \cos\theta_{(N-M-1)(m+1)} \end{bmatrix} = \begin{bmatrix} d_{1(m+1)} \\ d_{2(m+1)} \\ \vdots \\ d_{(N-m-1)(m+1)} \end{bmatrix} \tag{6.7}$$

$$\mathbf{D}' = [\mathbf{D}_k \ \mathbf{D}_{new}] = \begin{bmatrix} d_{11} & \cdots & d_{1(m)} & d_{1(m+1)} \\ d_{21} & \cdots & d_{2(m)} & d_{2(m+1)} \\ \vdots & \ddots & \vdots & \vdots \\ d_{(N-m)1} & \cdots & d_{(N-m)(m)} & d_{(N-m-1)(m+1)} \end{bmatrix} \tag{6.8}$$

Therefore, the new redundancy values can be computed by adding the new cost:

$$J_i' = J_i + \left(2d_{i(m+1)}^2 - 1\right) \tag{6.9}$$

To better understand on how the algorithm works, an example scenario is given in APPENDIX B.1.

6.2.3 Experimentation

MCSSA was validated by comparing its performance against the optimal algorithm (OSSA) and the three heuristic algorithms: MVA, QOA, and HEA. The performances were measured by positioning accuracy and computation time. To facilitate the analysis of positional accuracy, two forms of DOP values were considered: raw DOP values and DOP ratios ($\xi_{method} = \frac{DOP_{algorithm}}{DOP_{optimal}}$).

One thousand sample cases, representing user's positions, were randomly selected on the Earth's surface at a random time within one day, August 12, 2007. The GSSF software was used to determine potential satellites at each testing location with the cutoff elevation at 5°. All

algorithms were to select a set of satellites from the combined constellations composed of GPS and Galileo with a varying number of selected satellites from four to fourteen. On average, each location can observe about 21 satellites for the integrated constellation (GPS+Galileo) and 10 satellites for a single constellation (GPS or Galileo), see Figure 6.2.

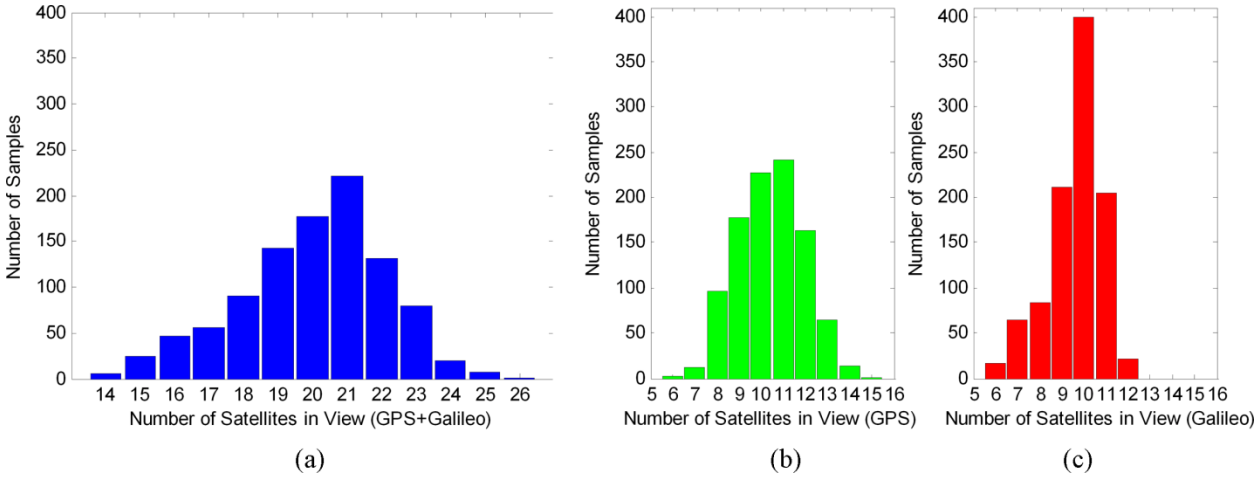
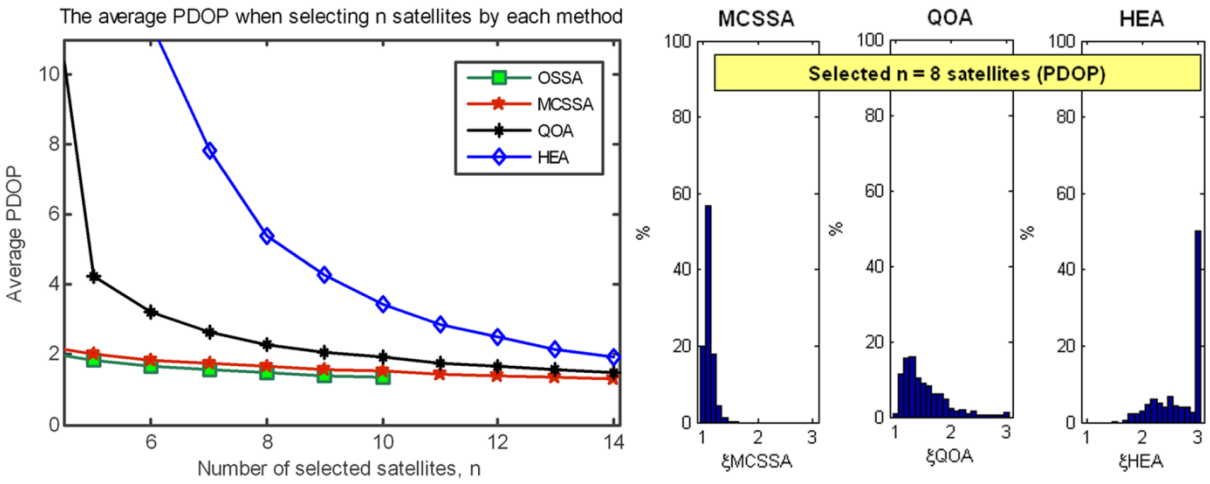


Figure 6.2. The number of visible satellites when operating (a) the combined systems (GPS+Galileo), (b) GPS, (c) Galileo for 1,000 cases with the cutoff elevation angle at 5°

6.2.3.1 Results and Discussion

Accuracy performance, in terms of PDOP, showed that MCSSA provided a set of selected satellites with PDOP near-optimal. Figure 6.3(a) shows the raw PDOPs produced by each algorithm. The PDOP values of MCSSA, on average, were about 10% higher than optimal solutions when varying the number of selected satellites. MCSSA was significantly better than the other two heuristic algorithms when a number of selected satellites were small. In addition, the trends of OSSA and MCSSA were nearly flat when the number of selected satellites increased; this means that the PDOP values improved less than 5%. The flat trend indicates the closeness to the minimum PDOP when all satellites were used in the calculation. The PDOPs given by OSSA and MCSSA were flat when selecting more than 8 satellites. Therefore, using 8 satellites selected by MCSSA can provide a near-optimal solution. Figure 6.3(b) shows the PDOP ratio when selecting 8 satellites. MCSSA provided ξ_{MCSSA} close to 1, indicating that its

solutions were almost always close to the optimal solutions, and always less than 2, guaranteeing that its solutions were not more than twice the optimal solutions.



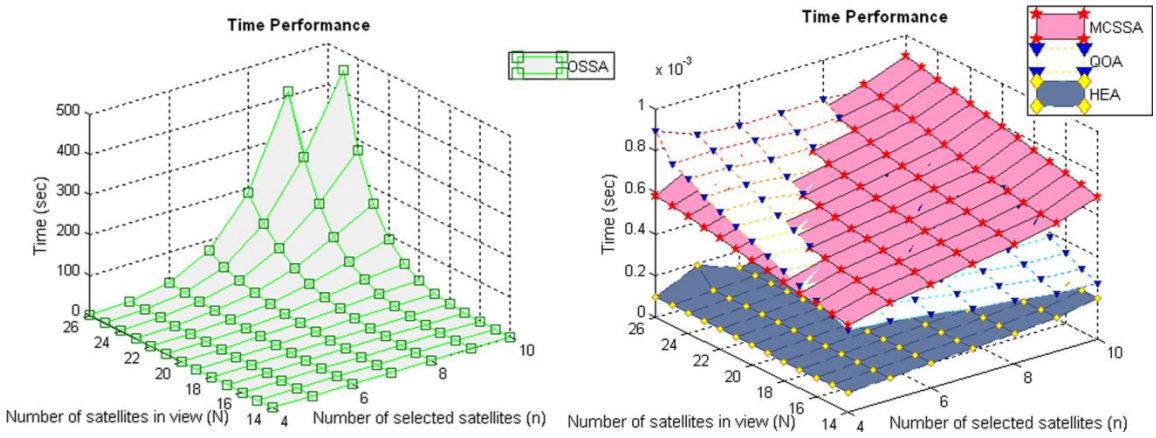
(a) Raw PDOP (b) PDOP ratio

Figure 6.3. Accuracy performances of satellite selection algorithms

The average time performances of each algorithm were measured when run with varying number of selected satellites. The optimal algorithm (OSSA) required significantly higher processing time than the heuristic algorithms and considerably increased in time when the number of selected satellites increased, as depicted in Figure 6.4(a). On the other hand, the heuristic algorithms required much shorter computation than OSSA. Comparing among the heuristic algorithms, HEA required the shortest computing time, see Figure 6.4(b). MCSSA performed better than QOA when selecting a small number of satellites from a large set of potential satellites. QOA performed better than MCSSA when the number of selected satellites increased. Due to minor differences in a fraction of milliseconds, MCSSA requires low computation time, which was comparable to the other two heuristic algorithms and greatly outperforms the optimal algorithm.

In conclusion, MCSSA can provide a set of selected satellites with DOPs near-optimal even with a small number of selected satellites (i.e., 8 satellites). In addition, MCSSA

significantly reduces computation time and is comparable to the computation time of the other heuristic algorithms.



(a) The optimal algorithm (b) The heuristic algorithms

Figure 6.4. Time performances of the testing satellite selection algorithms

6.3 GRID-BASED SIMULATION FOR IGNSS QOS PREDICTION

Taking a sequential computing approach for iGNSS QoS prediction, due to large size TIN datasets and a large number of satellites in iGNSS, may not provide real-time response. This section discusses the exploration of a grid computing platform with the objective of reducing computation time of iGNSS QoS prediction to meet the real-time requirement of navigation systems/services. Since calculating visibility of each satellite in the point-based prediction module is the most computationally expensive process, we conducted an experiment to understand performances and limitations of a grid computing platform (PittGrid) for satellite visibility calculation, i.e., LOS determination between the receiver and each satellite at a particular location and time. The process of examining a LOS, described in Section 4.2.4.1, is simplified in this section, called iGNSS-v, to reduce complexity while promoting a better understanding on the grid performance.

6.3.1 iGNSS-v

iGNSS-v computes whether a direct LOS between user's location and a satellite exists. iGNSS-v is different from the algorithm described in Section 4.2.4.1 in that it does not apply the 3-zone LOS calculation which uses different granularity of TINs. Instead, iGNSS-v determines LOS using only the high-resolution TIN data layer, see Figure 4.3 (a), to evaluate the worst computation requirement. The steps of iGNSS-v are: (1) compute a set of potential visible satellites, given user's location and time and (2) compute the intersection of direct LOS with the objects on the Earth's surface. The iGNSS-v algorithm using high-resolution TINs is presented in Figure 6.5.

The input parameters depending on location and time are user's location (*user_loc*), user's time (*user_time*), and almanac file(s) (*almanac*) while others, i.e., TIN data (*tin*), elevation cutoff angle (*ele_cutoff*), distance of truncated LOS (*l*), and 2D buffer width (*w*), are the pre-configurable parameters of iGNSS-v.

The first two steps determine potential visible satellites available at the user's location. Given a satellite almanac file(s) and an expected user's time, the position of all satellites can be approximated by the satellite motion model (e.g., the model for GPS is the Kepler model), expressed as *sat_pos*. Then, topocentric coordinates of each satellite at the user's location are computed and a set of potential visible satellites can be determined using a horizontal elevation cutoff relative to the user's location.

Instead of checking the intersection of the actual LOS with all objects on the surface, iGNSS-v reduces the search space by truncating the LOS to l km, projecting the LOS to a 2D horizontal plane, and retrieving the objects (triangles) that overlap with a window clip $l \cdot \cos\theta \times 2w$, as depicted in Figure 6.6 where θ is the incident angle of the LOS. With this, the number of triangles considered depends on the size of the window clip. Each triangle is checked for intersection with the actual LOS. The algorithm terminates upon interference detection.

```
Input Parameters:    user_loc, user_time, almanac, tin, ele_cutoff, l, w
Output Parameters:   los_visibility
iGNSS-v (user_loc, user_time, almanac, tin, ele_cutoff,  l, w){
   // determine satellite positions at a given time
   s ← sat_pos(user_time, almanac)
   // determine the potential visible satellites at the user's location
   ps ← potential_sat(s, user_loc, ele_cutoff)
   // the following statements determine visibility of each satellite
   m ← ps.length
   for i = 1 to m do {
            LOS_3D ← los_truncate(user_loc, ps[i].loc, l)
            LOS_2D ← project(LOS3D)
            2Dbuffer ← buffer( LOS_2D, w)
            tri ← window(tin, 2Dbuffer)
            n ← tri.length
            for j = 1 to n do{
               if (intersect(LOS3D, tri[j])) then
                     return invisible
               else
                     continue
            }
            return visible
   }
}
```

Figure 6.5. iGNSS-v algorithm for grid-based simulation

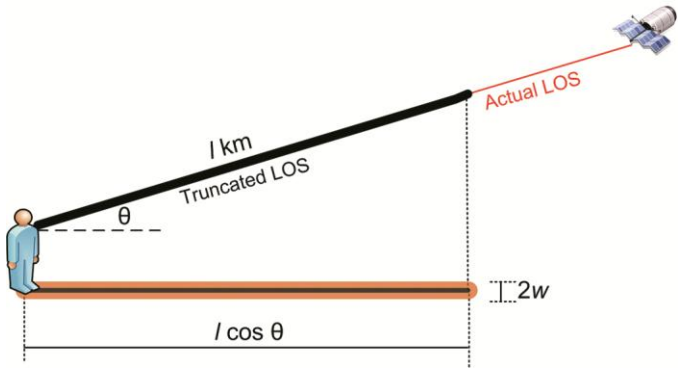


Figure 6.6. A truncated LOS and the window clip for iGNSS-v

While iGNSS-v is conceptually straightforward, it involves several computational issues. First, it involves large amounts of 3D data needed for realistically representing the environment. The correctness of iGNSS-v depends on how accurate the local environment is represented. The higher the resolution and details of the terrain and 3D objects, the longer it takes to retrieve surface data and execute visibility calculation. In addition, length of LOS affects the time needed to determine visibility. The actual distance between a position on the Earth's surface to any GNSS satellite is more than 20,000 km. Considering all objects along a 20,000-km LOS would require large amounts of data which are unnecessary since potential obstacles are most likely located only on the Earth's surface and not above. A truncated LOS from the user's position to the satellite, a virtual LOS, can reduce the amount of surface data to be considered. However, the issue of how long a virtual LOS should be is an open question. The longer the virtual LOS is, the more precise the iGNSS-v results will be at the cost of a higher computation time.

Second, the computation time of iGNSS-v at a given location and time increase linearly with additional satellites in the combined constellations. Currently, a GPS receiver can observe up to 12 satellites at any location and time but, in the future, an iGNSS receiver, utilizing all four GNSSs, could observe at least 50 satellites simultaneously. Thus, iGNSS-v for the four constellations requires about four times more computation time than a single GNSS.

Third, iGNSS-v is an iterative process in that, given a location and a time, iGNSS-v needs to determine visibility to all potential satellites. Sequentially executing this task would prolong the overall computation time needed to predict iGNSS QoS. In addition, given a route input iGNSS QoS prediction must compute QoS at multiple points and varying times. Sequentially executing these points for visibility calculations would significantly increase the computation time to provide iGNSS QoS prediction to applications.

Last, due to real-time constraint imposed by navigation systems/services, the acceptable window for waiting on a response is very short. In order to provide the predictive information in a timely manner, iGNSS QoS prediction must have enough computing power at its disposal to execute all the tasks, including iGNSS-v which is a dominant one, within a very short time. If a delay occurs, the predictive information may become useless.

Such computational and storage issues make current mobile receivers/devices unsuitable for iGNSS-v calculation. These limitations prohibit typical GNSS receivers from realistically predicting GNSS QoS at prospective locations. To that end, we explored parallelization of the

visibility calculation (the second step of iGNSS-v) using a grid computing platform; this is possible because satellites are processed independent from each other. It is expected that grid computing can scale up as the demand on the system increases.

6.3.2 Grid Computing

A grid computing platform is a group of geographically dispersed computers which are coupled via a network to provide multiprocessing capabilities (Foster et al., 2001). With the huge number of computing and data resources available, computational grids have proven to be an efficient execution infrastructure for a wide spectrum of compute-intensive applications benefiting fields as diverse as high energy physics, bioinformatics, astronomy, earth and life sciences, and finance (Berlich et al., 2005, Reed, 2003, Stevens et al., 2003). For details on grid computing technologies, techniques, and applications refer to Berman et al. (2003) and Liu (2005).

A grid computing platform in the University of Pittsburgh's main campus, called PittGrid[9], which has been operational since 2006 was used to experiment computation of iGNSS QoS prediction. PittGrid currently incorporates approximately 400 CPUs from departments and laboratories across the University of Pittsburgh's campus including those in Computer Science, Physics and Astronomy, School of Information Sciences, Mathematics, School of Nursing, Department of Chemistry, Department of Chemical Engineering, Department of Statistics, and the Freshman Engineering Laboratory. PittGrid provides high-throughput computing and high-performance computing for faculty, researchers and students to carry out computational and simulation tasks as well as to run computationally demanding projects.

PittGrid creates a virtual supercomputer by harnessing the unused CPU cycles across the campus, based on the Globus grid middleware from Globus Alliance, an open source toolkit to provide grid computing functionalities. It also incorporates Condor, open source software from the University of Wisconsin, which uses heterogeneous computing resources and the ClassAd mechanism to match the submitted jobs to the available computing resources. PittGrid uses Open Source Grid Services Architecture for Data Access and Integration (OGSA-DAI) to provide access to different databases in the environment. It supports both serial and parallel jobs. Some

[9] http://www.pittgrid.pitt.edu

of the PittGrid nodes constitute the virtual parallel computing cluster and can run Message Passing Interface (MPI) tasks. PittGrid supports FORTRAN, C, C++, and Java. It also supports special software such as MATLAB and R, Stata (Statistics software) and Amber (Chemistry software).

6.3.3 Experimentation

Two sets of experiments were conducted to study the feasibility of using PittGrid for iGNSS-v (Karimi et al., 2011c). The first experiment used fully distributed data over database nodes with a varying number of clients between runs to compute in parallel. The second set of experiments used fully replicated data over the same database nodes with a varying number of clients between runs as well. Only GPS satellite constellation was considered in this study.

Four point cloud tiles, covering the downtown area of Pittsburgh and the University of Pittsburgh's campus (tile index = 41001340PAS, 42001340PAS, 41001350PAS, 42001350PAS in Figure A.1), were used for constructing four TINs in Oracle. Four Oracle Database nodes were set up in four different machines. Four client nodes were established to carry out LOS calculation. In order to test the effect of distributed computation, the number of clients was varied in both experiments. Both experiments also used a Job Scheduler node to divide up the LOS visibility determination evenly among the client nodes and decide which database node(s) each client's job would communicate with. All machines used in the experiments were of the same hardware specifications: Intel Core 2 Duo processors rated at 2.13 GHz, and 2GB of RAM. The OS on all machines was Microsoft Windows Vista Ultimate.

The two experiments were carried out using three different receiver locations at the same observation time, which assured the same set of satellites above the horizon. Figure 6.7 shows the three positions: center, the center of northeast tile (quadrant B), and the center of southwest tile (quadrant C).

Two different data distributions were used in the experiments: fully distributed and fully replicated. In the fully distributed case, each of the four TINs was stored on a separate database node. The individual databases were complementary to each other when providing the entire data set. In the fully replicated data distribution, all four TINs were stored on four database nodes.

Figure 6.8 shows the fully distributed case and the fully replicated data distribution within the grid.

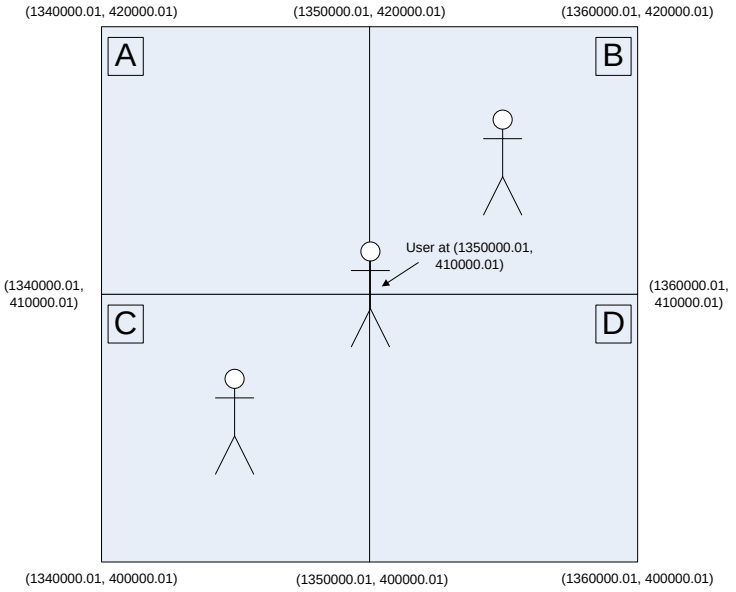
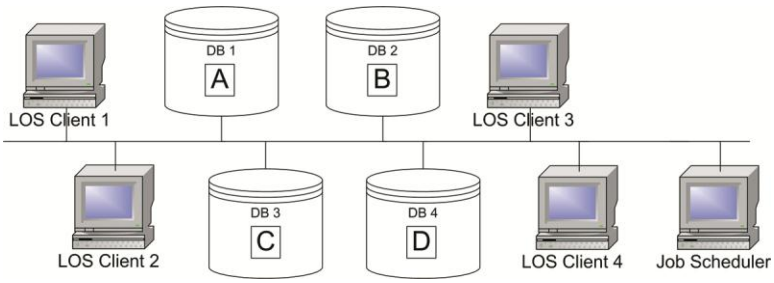
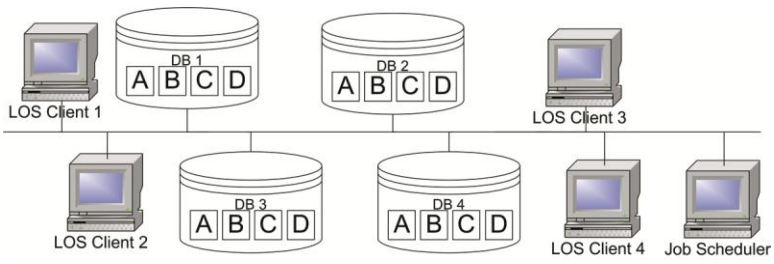


Figure 6.7. Three testing locations for grid-based simulation



(a) Fully distributed



(b) Fully replicated

Figure 6.8. Data distribution in the grid

For the distribution of iGNSS-v, the process was divided into two parts. The first part, or the first-step of iGNSS-v, was assigned to the Job Scheduler node while the second step of iGNSS-v was assigned to client nodes. The process of each experiment run is presented in Figure 6.9. First, user's location and time were submitted to the Job Scheduler node of PittGrid. The Job Scheduler then determined the GPS satellites' positions. Next, all the database nodes known to the Job Scheduler were polled to determine the locations of all TIN datasets through the metadata containing the extent of each TIN. Note that the metadata was retrieved and stored at the Job Scheduler node for each user's location; therefore, it is a one-time retrieval from the database nodes.

The Job Scheduler then iterated through all satellites on user's horizon and determined which database node contained the data necessary for a LOS calculation between the user and the satellite. It then passed user's position (x,y,z), satellite's position (x,y,z), the data necessary for connecting to the database node including the host name, username, password, and the table name which contained the TIN data on to a client machine. Each individual client made a specific query on TIN data (i.e., all triangles within a given LOS path) in the given database node and determined the visibility of the given satellite based on the queried TIN data.

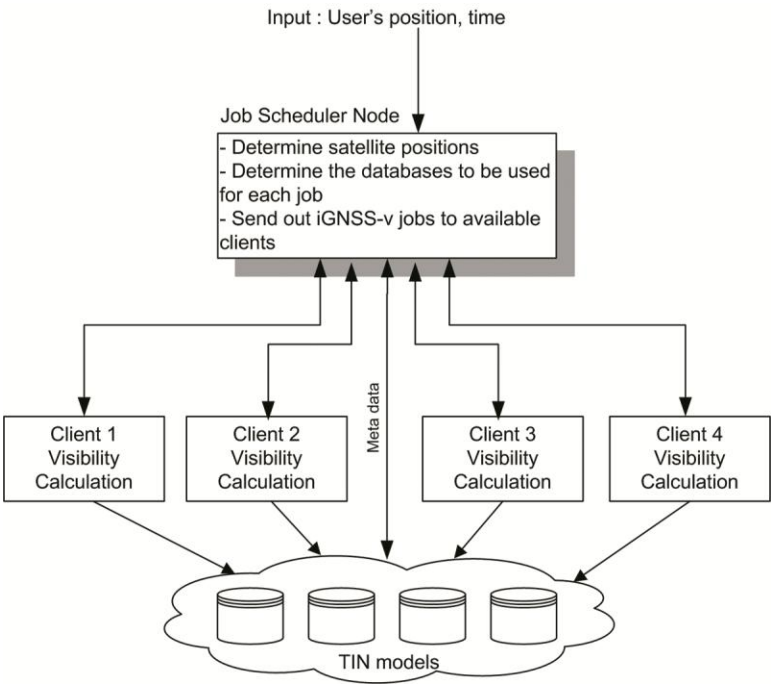


Figure 6.9. Distributed iGNSS-v in the grid

6.3.3.1 Results and Discussion

Truncated Distance and Buffer Size Analysis

Before we analyze the results of the two experiments, we discuss the results of a test we conducted to realize the impact of varying size of truncated distance and buffer on computational time. This test was controlled with the following settings: user's location is at the center of quadrant C in Figure 6.7, specific time, and using the fully replicated data case. Thus, the impact of data distribution will not influence the computational time of this test. The number of LOS clients was varied from one to four nodes. The truncated distance (l) was set as 100 m, 1 km, and 10 km. The buffer size was set as 1 m, 3 m, and 5 m. Thus, the total of 36 cases was set and each case was run three times.

The average run time of each case is reported in Figure 6.10. It is clear that the size of truncated distance highly influences computation time of visibility calculations. Figure 6.11 shows the average run times caused by different truncated distances with the fixed buffer size (w = 1 m). We observed that the truncated distance has a linear relationship with computation time; meaning that increasing the size of truncated distance causes a linear increase in the computation time of visibility calculation. The truncated distance impacts not only the size of retrieved data but also preciseness of visibility calculation. However, in this test, no differences were found on the visibility results when varying truncated distances. We could not draw a conclusion on the accuracy of visibility results using different distances due to the use of a single location and time. In future research, different truncated sizes may be applied for different environment settings to optimize computation time while maintaining accurate visibility calculation. For example, within a city or urban area, a smaller truncated distance would be used since it is likely that the surrounding objects or buildings near user's location will block LOSs, as shown by the results in Section 4.4.4.2. In contrast, within a rural area a longer truncated distance may be necessary because it is likely that terrain or the Earth's surface will block LOSs.

The small variation of buffer size has minor impact on run times because buffer size is relatively small comparing to truncated distance ($w \ll l$) which causes slight increase in retrieved data. Only small buffer sizes were considered because we observed that the objects farther away from the projected line of the direct LOS have a smaller chance of blocking LOSs. Thus, retrieving such extra information from the surface data will not increase quality of visibility

solutions. Figure 6.12 shows a sample plot of the average run times with various buffer sizes and a fixed truncated distance (l = 1 km). We observed that using the 1-m buffer required slightly higher computation time. This could be due to the overhead of the Oracle database management functions, such as indexing and filtering.

In order to measure the performance of parallelization and data distribution in a grid platform, we used the worst case scenario that is l = 10 km and w = 1 m as the initial parameters for iGNSS-v.

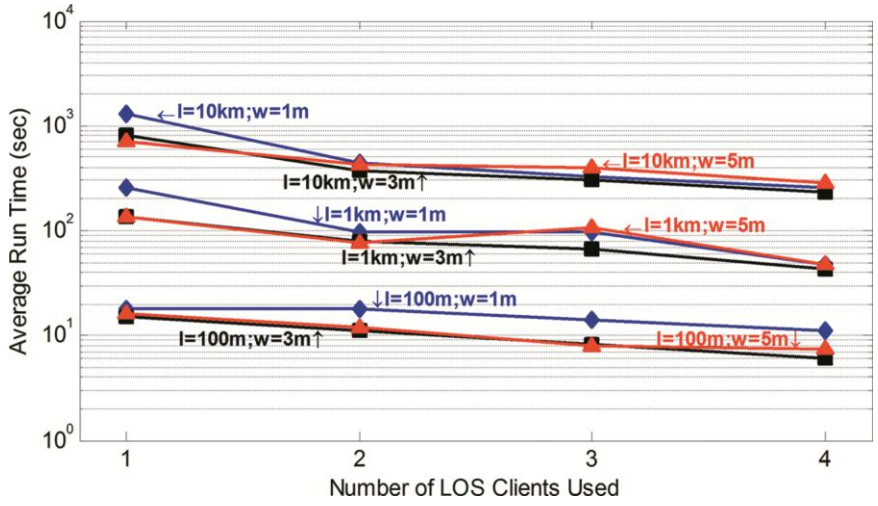


Figure 6.10. Average run times with varying sizes of truncated distance (l) and buffer (w)

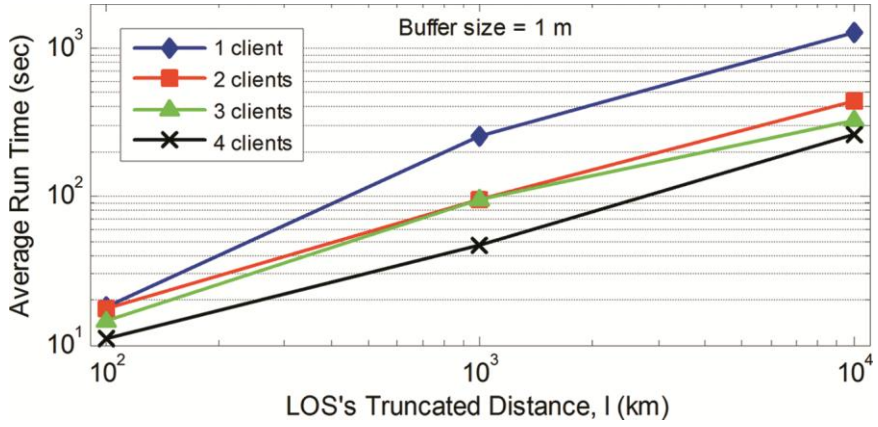


Figure 6.11. Relationship between LOS truncated distance (l) and average run time with buffer size (w) = 1 m

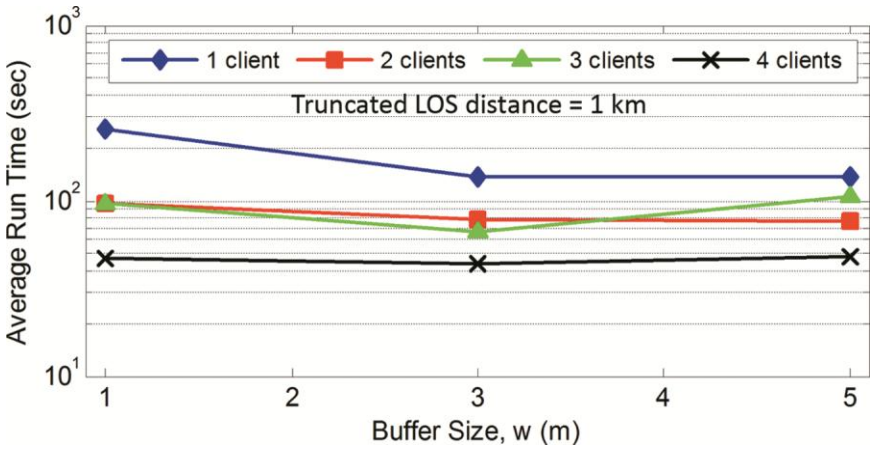


Figure 6.12. Relationship between buffer size (*w*) and average run time with truncated distance (*l*) = 1 km

Parallelization and Data Distribution Analysis

Figure 6.13 shows the total run time in the grid-based LOS simulations using fully distributed data and fully replicated data when varying the number of clients from one single client to four clients running in parallel to perform LOS calculation. The total run time is calculated as follows:

$$T = t_0 + t_1 + t_2 + \sum_{i=1}^{n} t_{3i} \quad (6.10)$$

where t_0 is the time to determine satellites' positions,

t_1 is the time to poll database nodes to determine the location of data,

t_2 is the time for the Job Scheduler to determine which databases each LOS job would use to compute visibility of each satellite,

t_{3i} is the time for each LOS job to execute visibility calculation,

n is the number of LOS jobs.

For each LOS job, t_3 can be broken down into three separate times as follows:

$$t_{3i} = t_{3a} + t_{3b} + t_{3c} \quad (6.11)$$

where t_{3a} is the time for database node to read its data from hard disk,

t_{3b} is the time for data transfer across the network,

t_{3c} is the time for the LOS client to carry out the intersect operation.

Of all these times, t_{3a} was found to be the dominant factor, accounting on average for greater than 99% of each job's execution time.

In the fully distributed case, the average run time for one client was 1,531 seconds (about 25.5 minutes). The run time improved according to the curve shown in Figure 6.13, reaching an average run time of 645 (about 11 minutes) when run over four clients in parallel, about 58% improvement in computation time. In the fully replicated case, the average run time for one client was 1,446 seconds (about 24 minutes). The run time decreased according to the curve shown, reaching an average run time of 353 seconds (about 6 minutes) when run over four clients in parallel, about 74% improvement in computation time. Using four clients can achieve a 45.2% improvement in run time when the fully replicated data distribution was used as opposed to fully distributed data distribution. A diminishing return with respect to run time decrease as additional clients were added was also displayed by both experiment settings, thus making the curve non-linear.

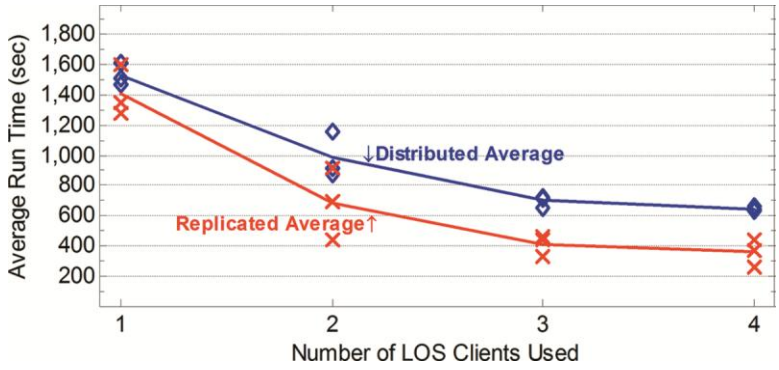


Figure 6.13. Average run times using fully distributed data and fully replicated data

The lower performance of the fully distributed data distribution can be explained by concurrent database accesses by multiple LOS clients. The Oracle database management system attempts to provide data in a parallel manner. However, this particular application is very data-intensive which severely reduces the ability of the Oracle software to provide data in parallel to two separate clients. Clients had to queue to access data. In the fully replicated data distribution,

all clients could communicate with a dedicated database node, which eliminated any queuing delays. All the data was available to each client whenever desired.

When different user's locations are compared for run times, the location in the northeast performed the slowest on average, the southwest position performed the fastest on average, and the central location performed in between the two. This is the case for both distributed data and replicated data. The southwest and northeast locations required more LOS jobs to be carried out in the distributed data case because, one LOS may cross several TIN datasets. Since the datasets were stored in separate database nodes, separate LOS jobs had to be created for each TIN dataset crossed. This was not the case in the fully replicated data distribution, where each LOS took only one job, because every LOS could be queried within one database node.

The amount of data storage increased greatly between the two data distributions. In the fully distributed case, a total of 2,400 MB (4 sets × 600MB/set) of storage was required to store the TIN datasets. In the fully replicated data distribution, a total of 9,600 MB (4 database nodes × 4 sets/node = 600MB/set) of storage was required. This is a 300% increase in data storage. When four client nodes were used in parallel, a 58% improvement in run time was achieved when switching from the fully distributed data to fully replicated data.

Given that the client nodes are shared resources within the grid, a smarter scheduling scheme would be to consider the current load on the nodes and schedule according to least CPU usage in the nodes. Another possibility is to consider exclusive nodes for the purpose of iGNSS-v.

Although the current experiment setting does not provide real-time solutions to iGNSS QoS prediction, it has proven that a grid is a feasible platform for reducing computation time of visibility calculation.

6.4 CLOUD-BASED SIMULATION FOR IGNSS QOS PREDICTION

The feasibility of a cloud computing platform to store and manage large TINs was explored. Google App Engine (GAE) was chosen due to the fact that currently it is the only cloud computing platform available to researchers at no cost. Given that iGNSS QoS prediction requires geospatial data indexing and processing, GeoModel, which is an open source library for

the GAE datastore, was employed. Two experiments were conducted to evaluate the performance of GAE and GeoModel on point data and TIN data, respectively, to perform proximity and bounding box queries (Karimi et al., 2011b).

6.4.1 Cloud Computing

Cloud computing is now widely viewed as a promising paradigm for building tomorrow's geoprocessing systems (Brauner et al., 2009). Several projects both in academia and in industry have recently started efforts to develop prototypes of geospatial applications on clouds. Cornillon (2009) explored the suitability of cloud computing for processing large volumes of satellite-derived sea surface temperature data. Hill (2009) presented the results of experiments using Amazon's Elastic Compute Cloud (EC2) for ocean-atmosphere modeling. Blower (2010) presented an implementation of a Web Map Service (WMS) for raster imagery using the GAE environment. Wang (2009) described a prototype for retrieving and indexing spatial data developed for GAE.

In parallel with these efforts in academia, several GIS vendors have recognized the promise of cloud computing and some have already introduced cloud-based GISs. ESRI provides preconfigured ArcGIS Server Machine Images (AMI) for use in the Amazon Cloud infrastructure (ESRI, 2010). Running ArcGIS Server on Amazon allows organizations to deploy ArcGIS Server across multiple data centers and access the Amazon's elastic computing infrastructure. GIS Cloud (Omnisdata, 2010) is a Web-based GIS powered by cloud computing with advanced capability for creating, editing, uploading, sharing, publishing, processing and analyzing geospatial and attribute data. Kim and Mackenzie (2009) used Amazon's EC2 for climate change study with the purpose of calculating the number of days with rain in a given month on a global scale over the next 100 years. The study used 70 GB of daily sets of climate projection data and took about 32 hours to process 17 billion records. In the absence of spatial data and processing in Windows Azure, ESRI's MapIt, which features Spatial Data Assistant (SDA) and Spatial Data Service (SDS), is used (ESRI, 2009).

Cloud computing is seen as the needed paradigm to finally shift the (often intensive) processing part of geospatial applications from the desktop to distributed spatial data infrastructures (Schäffer and Baranski, 2009, Yang et al., 2010). By outsourcing data- and/or

compute-intensive tasks to clouds, geospatial applications will benefit in terms of performance, scalability, and start-up cost.

While most initial research findings have concluded that cloud computing is a viable paradigm for data- and compute-intensive geoprocessing tasks, the fundamental limitation that current cloud infrastructures are for generic computing still remains. Furthermore, cloud infrastructures often are not aware of the spatial nature of the data. As a result, current cloud computing infrastructures still require extensive improvements and extensions to become geospatial clouds.

6.4.2 Experimentation

To explore the capabilities and performance of a cloud computing platform for data-intensive geoprocessing tasks, we focused on the TIN data queries required by iGNSS QoS prediction as a case study and used GAE as a cloud computing platform. Of the existing clouds, which have different characteristics, functionalities, limitations, and requirements, GAE was chosen due to its generic and full featured platform that allows developers to test their web applications on a cloud platform in a short time and its publicly available service at no (or low) cost. Other choices included geospatial-oriented software in clouds, such as ESRI products, but they were not chosen due to their proprietary data structures and formats. Since the current GAE platform does not support geospatial data and processing, GeoModel was used to index geospatial data and perform basic spatial operations (i.e., proximity and bounding box). For this reason, before discussing and analyzing our evaluation of GAE for handling iGNSS QoS prediction, the performance of GAE's datastore with the use of GeoModel to store TIN data and the performances of proximity and bounding box operations are analyzed. Figure 6.14 illustrates an architecture for deploying geospatial applications in GAE with GeoModel.

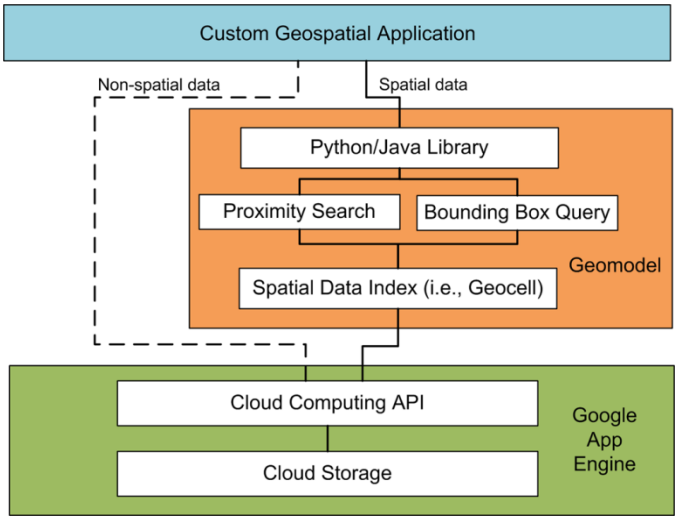


Figure 6.14. An architecture for geospatial applications based on GAE and GeoModel

6.4.2.1 Google App Engine

GAE is a platform as a service (PaaS) cloud computing environment which provides services for developing and hosting web applications with low implementation overheads. Many system administrations and challenges for developing a scalable web application have been addressed by GAE[10]. Developers do not need to maintain servers or reserve resources (e.g., CPU, machines, and storage spaces) in advance. Once the application is deployed in GAE, GAE will scale instances of the application as its demand or load grows. As a result, web developers can build a large-scale, data- and/or compute-intensive applications with ease and in a short time. Thus, GAE is utilized as a black box where issues of data distribution and scalability are automatically handled by the platform. However this, in turn, provides the web developers with no control over the behavior of the GAE environment.

With automatic processes at GAE's backend, Google allows developers to use only supported languages (i.e., Java and Python), APIs (e.g., for storing and retrieving data from Bigtable – Google's non-relational database, for making HTTP requests, and for sending e-mails), and frameworks (e.g., Python web frameworks). In addition, there are several restrictions such as no files can be written to the GAE's file system, only a subset of classes in the standard

[10] http://code.google.com/appengine/docs/whatisgoogleappengine.html

Java Development Kit are available, spawning background threads are not allowed, and the amount of random-access memory available to each service instance is low (Blower, 2010).

To store and retrieve distributed data efficiently in GAE, Google provides the datastore which is a distributed data storage service that performs distribution, replication, and load-balancing automatically. The size of the GAE datastore is scalable with application's data. The GAE datastore is built on two Google services: Google File System (GFS) and Bigtable (Roche et al., 2009). These two services are proprietary technologies for dealing mainly with scalability and distribution issues for large-scale data. GFS is a distributed file system (Ghemawat et al., 2003) while Bigtable is a compressed and high performance database system (Chang et al., 2008). With the distributed nature of Bigtable and GFS, the GAE datastore is non-relational database with no schema. All data are stored as data objects, or entities. Each entity has a unique identification and a set of properties. The GAE datastore provides full CRUD (create, read, update, and delete) operations to access entities in Bigtable by using a SQL-like language called GQL. The major differences between SQL and GQL is that GQL does not support joins and imposes several restrictions on a query, such as only one inequality filter ($<, \leq, >, \geq, \neq$), are allowed on a property, and filtering or sorting on a property requires that the property exists[11]. With these limitations of GQL including no joins, lack of cause statement, and lack of complete inequality on a property filtering, geospatial queries cannot be simply created in the GAE environment.

GAE offers free or billing service for each developer account with different quotas and limits on the used resources. Fees are charged to the billing service for additional usage of resources that are over the quotas of free services[12]. Some quotas are reset every 24 hours such as CPU Time and some are fixed quotas such as the total stored data.

6.4.2.2 GeoModel

GeoModel is an open source project that provides basic indexing and querying of geospatial data in the GAE environment (Nurik and Shen 2009). GeoModel indexes geospatial entities, currently only single points having latitude and longitude, by using a concept called "geocell". Geocell is a hierarchical structure that divides latitude and longitude space into

[11] http://code.google.com/appengine/docs/java/datastore/queriesandindexes.html#Restrictions_on_Queries
[12] http://code.google.com/appengine/docs/quotas.html.

rectangular grid cells (Nurik and Shen, 2009). The geocell generation process initially starts with the lowest resolution (1-digit string) by dividing the entire [-90, 90] x [-180, 180] space into the 4-by-4 rectangular grid. Each rectangular cell is labeled by a hexadecimal string, as shown in Figure 6.15. Subsequently, the process further divides each grid cell into 16 sub-rectangles for the next higher resolution. The process repeats until the maximum geocell string is reached (13-digit string). The geocell concept is similar to geohash[13] in the way that each digit in the prefix of each geocell string is referred to its ancestor.

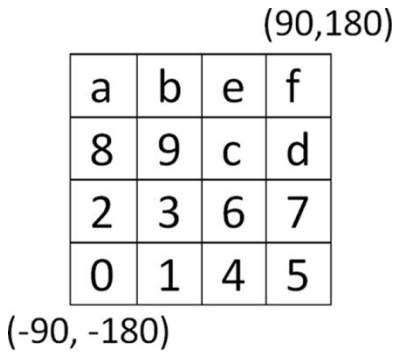


Figure 6.15. Hexadecimal strings assigned to geocells

For each point entity stored in GAE's datastore, GeoModel assigns a property called *geocells*, which contains a list of corresponding geocells from the lowest to highest resolution. For example, a point entity at latitude = 40.43813843 and longitude = -79.93741275 has a total of 13 items in the geocells property (9, 9a, 9a9, 9a97, 9a978, … , 9a97841f531eb).

Based upon the geocell concept, GeoModel currently provides two types of spatial queries: proximity and bounding box. Proximity query retrieves all point entities within a distance from a target location. Given a target location and a proximity distance, the process of proximity query has the following steps: (1) compute the highest-resolution geocell that contains the target location, (2) query the GAE datastore for all point entities that belong to the computed geocell, (3) calculate and sort the distance between each entity's location of the current geocell and the target location, (4) expand the search to four adjacent geocells or a lower resolution

[13] http://en.wikipedia.org/wiki/Geohash

geocell until it covers the pre-defined distance, and (5) return the entities within the pre-defined distance. Bounding box query retrieves all point entities within a pre-defined rectangular bounding box. The process of bounding box query starts by: (1) calculating the geocells that overlap with the bounding box, (2) querying the GAE datastore for all point entities belonging to the computed geocells, and (3) merging and returning the results of each geocell.

6.4.2.3 TIN in Cloud

A TIN is an approximation of a surface with a set of non-overlapping contiguous triangular facets of irregular size and shapes (Van Kreveld 1997). It is a vector model commonly used in GIS software to represent continuous fields such as terrain. Traditionally, a TIN is stored as a file, in ASCII or the ESRI TIN dataset files formats, which is not efficient for processing large TINs. There have been many attempts in designing TIN data structures and operations for spatial databases that allow storing, querying and reconstructing TINs across multi-scale databases efficiently (Al-Salami, 2009, Hanjiang et al., 2008). Currently, there are no standards on TIN data types and operations and there are no open source spatial databases that have features to efficiently store and manage TINs. Among commercially available spatial databases, Oracle has defined a proprietary data type and operations for managing large TINs. In this work, we designed a schemaless TIN model based on the data structure of Oracle's TIN suitable for GAE's datastore. The following sections describe Oracle's TIN

Oracle TIN Model

Oracle defines a scalable storage framework that can efficiently manage a large amount of TIN data. It provides various procedures for creating and querying TINs in a package called SDO_TIN_PKG (Kothuri et al., 2007). The default method for triangulation given a set of points is the Delaunay triangulation. The created TINs are stored in two tables: base table and block table. The base table stores a metadata describing each TIN, such as table name and extent. The points and triangles of each TIN are divided into blocks and stored as multiple rows in the block table. In order to maintain triangles and points of a contiguous region, TIN blocks are classified into at least two levels (*tr_lvl*). Each block with *tr_lvl* = 1 has no overlapping extent with other blocks at the same level. Each block with *tr_lvl* = 2 covers a pair of adjacent blocks with *tr_lvl* = 1. Thus, the blocks in *tr_lvl* = 2 contain triangles along the border of adjacent blocks in *tr_lvl* =1.

Sample triangles and points of blocks *tr_lvl =1* are shown in Figures 6.16 (b) and (c), blocks of *tr_lvl* = 2 are shown in Figures 6.16 (d) to (f), and Figure 6.16 (a) shows the region covered by the five blocks.

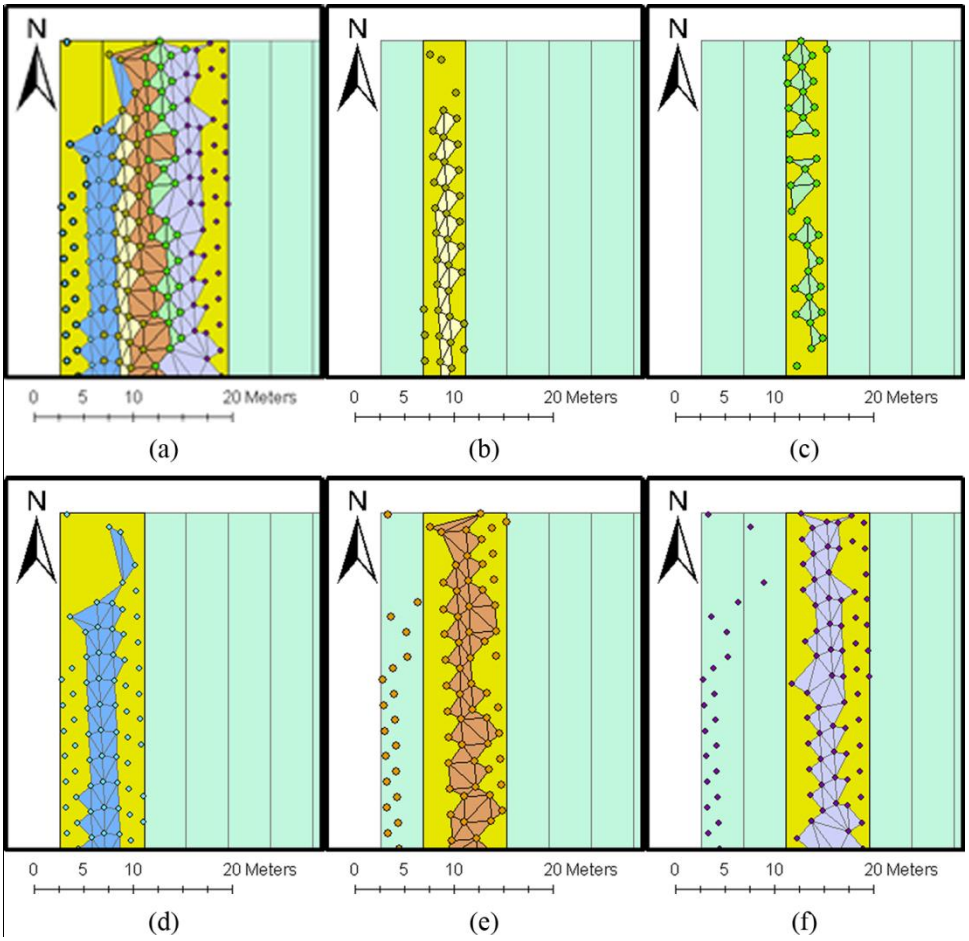


Figure 6.16. Sample Oracle TIN data stored as blocks: (a) total coverage area, (b) (c) blocks *tr_lvl* =1, (d), (e), and (f) blocks *tr_lvl* = 2

TIN Model for iGNSS QoS Prediction in Cloud

To maintain scalability of TINs created by Oracle database while accommodating the concept of schemaless entities in cloud computing, a TIN model appropriate for iGNSS QoS prediction was designed. The design is a variant of a triangle-based structure proposed by Poiker

(1990) and recapped for the database perspective by (Al-Salami, 2009). It composes of two types of data entities: *TIN_points* and *TIN_triangles*, as shown in Figure 6.17. Both types have unique keys that are encoded by concatenating strings of the Oracle block ID and the assigned point ID or triangle ID of that block. The *TIN_points* type has five properties: Latitude, Longitude, Altitude, Geocells, and TriangleList. The Geocells property contains a list of geocell strings of points from lowest to highest resolutions (1 to 13 digits). The TriangleList property contains a list of TIN_tringles keys that use points as their vertices. The *TIN_triangles* entity has two properties. The PointCoordinates property contains a list of coordinates (Latitude, Longitude, Altitude) for the three vertices of each triangle. The PointsID property contains a list of *TIN_points* keys associated with each triangle.

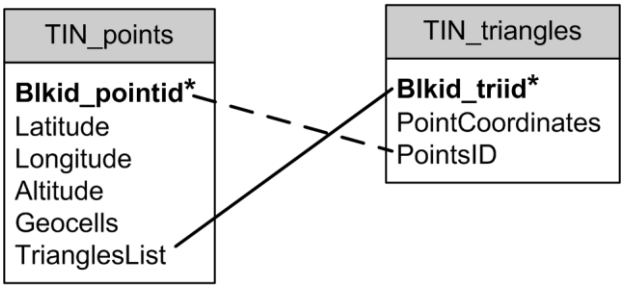


Figure 6.17. A TIN model for GAE datastore (* indicates a key)

Querying TIN

To query triangles from the GAE datastore using GeoModel, a two-step process is required. The first step is to retrieve the *TIN_points* entities for points that fall within the querying area through the use of GeoModel's operations. Once the points are retrieved, all triangles associated with them are extracted from the *TriangleList* property. The second step is to retrieve the *TIN_triangles* entities given a list of triangle keys. With this two-step process, a TIN that covers the querying area is reconstructed from the coordinates of returned triangles.

6.4.2.4 Method

Two experiments were conducted using the Java runtime environment to accommodate the TIN model discussed in Section 6.4.2.3. The codes were uploaded and run on GAE. The study area covers the University of Pittsburgh's main campus and the surrounding neighborhoods with a 3.048 km by 3.048 km area as shown in Figure 6.19(a). A TIN of this area was created from LiDAR point cloud with a point spacing of 1 m by using Oracle Database 11g. The total number of LiDAR points for this area is about 3.4 million. The TIN was stored in 1,371 blocks or records in the block table (686 blocks for *tr_lvl* =1 and 685 blocks for *tr_lvl* =2). The total number of triangles and points are about 6.8 million and 10.2 million, respectively. However, these numbers include redundant points and triangles. To reduce storage space, only points that are vertices of triangles were uploaded to the GAE datastore.

Experiment I focused on realizing the performance of GAE and GeoModel on point data with a varying number of point entities stored in the GAE datastore. The steps are: (1) incrementally uploading *TIN_points* to the GAE datastore (100 entities at a time), (2) performing proximity query and measuring the elapsed time at the client side for response, and (3) performing bounding box query and measuring the elapsed time. To better understand the worst case scenario, both types of queries were set to retrieve all entities from the datastore.

Experiment II focused on realizing the feasibility of using GAE and GeoModel to store TINs to allow visibility calculations for real-time iGNSS QoS prediction. The steps are: (1) uploading both *TIN_triangles* and *TIN_points* of the study area (up to the limit of the quota), (2) performing proximity query with different proximity distances and measuring elapsed time for response, (3) performing bounding box query with different sizes and measuring elapsed time, and (4) validating the returned results. Both types of queries were set to retrieve points and triangles.

6.4.2.5 Results and Discussion

The performance results are presented in Figure 6.18 for Experiment I. A majority of run times for both query types was spent on searching point entities within the computed geocells. Since the geocells property is not the entity key for points, a strong linear behavior when searching through geocells can be observed, R^2 = 0.9487 (proximity query) and R^2 = 0.8791 (bounding box query). In addition, proximity query took longer than bounding box query, mainly

because proximity query involves expensive computation on calculating and sorting the relative distance between each point entity and the target point, which is an iterative process inside in the proximity operation, and not parallelized in the GAE cloud. As the underlying infrastructure does not support spatial search capabilities or allow multiple entity keys, it is difficult at the application level, like GeoModel, to program an optimal code for expediting the geoprocessing tasks. In addition, it also can be observed that the elapsed time is unstable due to the nature of sharing resources among applications in GAE, which is seen as one of the critical obstacles to cloud computing (Armbrust et al., 2009). The elapsed time exceeded the 30-second limitation when the number of points reached 5,800 in proximity query.

According to these observed trends, the maximum point entities that can be retrieved from the GAE datastore through the use of GeoModel is about 6,700 and 14,392 entities for proximity query and bounding box query, respectively, under the limit of 30 seconds per response.

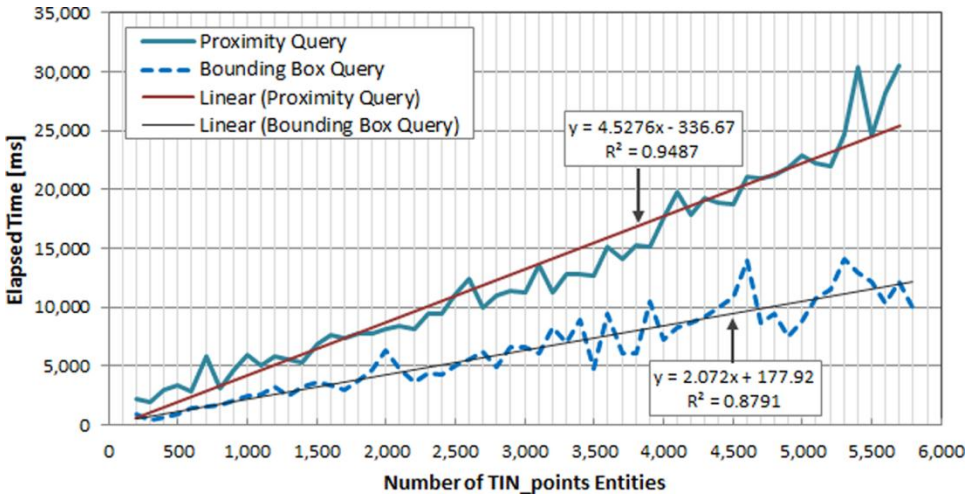


Figure 6.18. Time performance for retrieving the entire point entities in Google App Engine

For Experiment II, the Oracle TIN triangles and points were planned to be uploaded to the GAE datastore. However, due to the limited quota on the total storage space (1 GB), the triangles and points from 43 contiguous Oracle TIN blocks (22 blocks at *tr_lvl* = 1 and 21 blocks at *tr_lvl* = 2) were uploaded to the GAE datastore, which reached 100% of the storage quota. The

shaded area on the right of Figure 6.19(a) is the coverage area of the uploaded blocks (about 100 m x 3,048 m) and Figure 6.19(b) shows the boundary of those TIN blocks. A total of 225,369 points and 226,120 triangles could be uploaded.

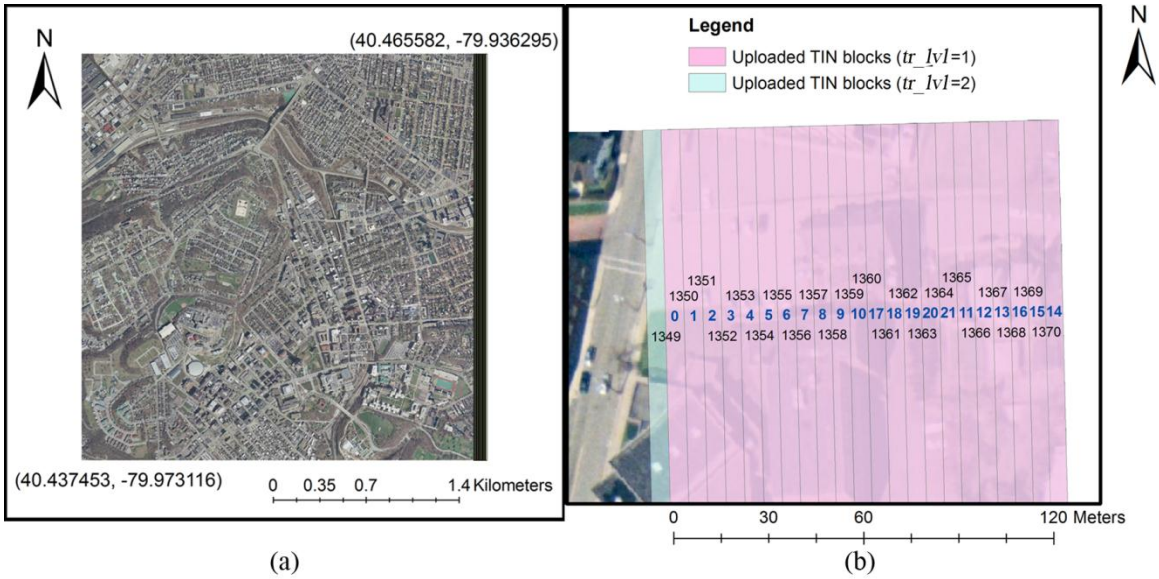


Figure 6.19. Extent of Oracle TIN's block: (a) study area and (b) uploaded data

Proximity query was performed at ten sample locations with the proximity distances of 10, 20, 30 and 40 m, as shown in Figure 6.20. The proximity distance of 40 m around user's location is sufficient to determine possible objects that potentially cause signal reflection, as described in reflected LOS calculation in Section 4.2.4.3. The performance results are presented in Figure 6.21. Only 32 queries were executed within the limit of 30 seconds. No result was returned for proximity query of 20, 30, and 40 m at Locations 1 and 4 and none for 40 m at Locations 2 and 3. Most of the run time was on retrieving points, the first step of TIN query. Increasing the proximity distance causes GeoModel to examine more relevant geocells, retrieve point entities of the computed geocells from the GAE datastore, and calculate distance and sort the retrieved entities by distance. This process added to the latency of the results, as found in Experiment I. The elapsed times varied greatly especially for smaller proximity distances. This

may be because Experiment II has more points in the GAE datastore which cause GAE to search point entities of the computed geocells from different physical storages.

While the process of retrieving triangle entities, the second step of TIN query, was relatively fast, on average, it took a constant time about 1, 3, 5, 9 seconds for the proximity distance from 10 to 40 m, respectively, after the point entities have been fetched. The process for triangle retrieval was significantly faster than point retrieval since the triangles were directly queried from the GAE datastore using the triangle entities key, with no GeoModel operations. To validate the results, Figure 6.22 shows sample plots of proximity query results. The points and triangles entities within the specified area for all testing distances were retrieved correctly and completely.

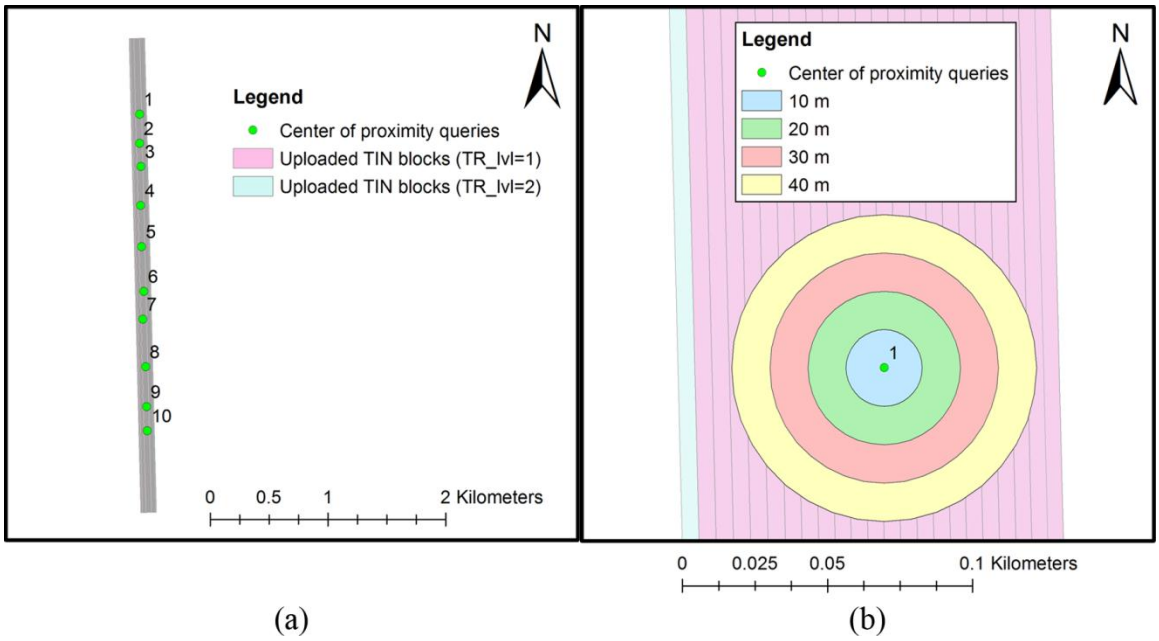


Figure 6.20. Proximity query: (a) ten sample locations and (b) proximity distances.

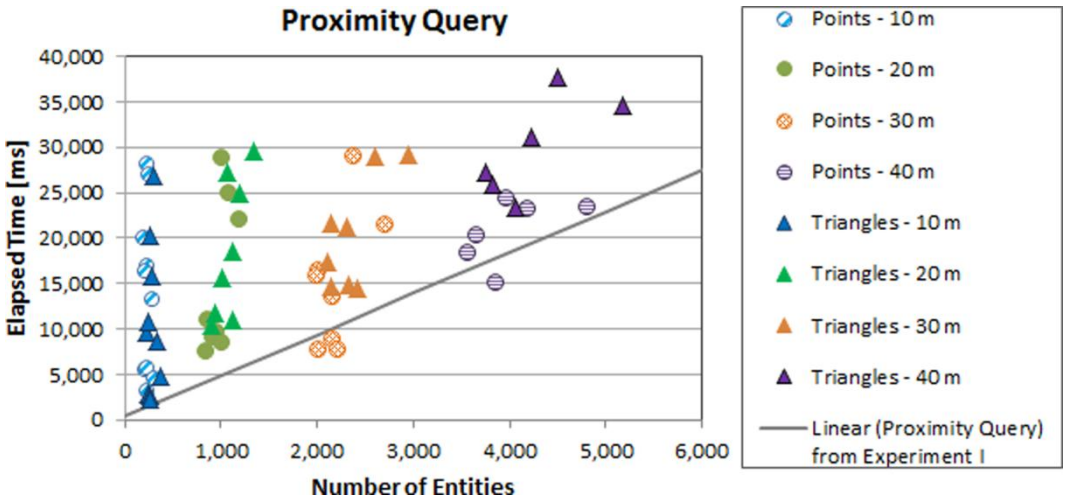


Figure 6.21. Time performance for retrieving TIN data, points and triangles, for proximity query

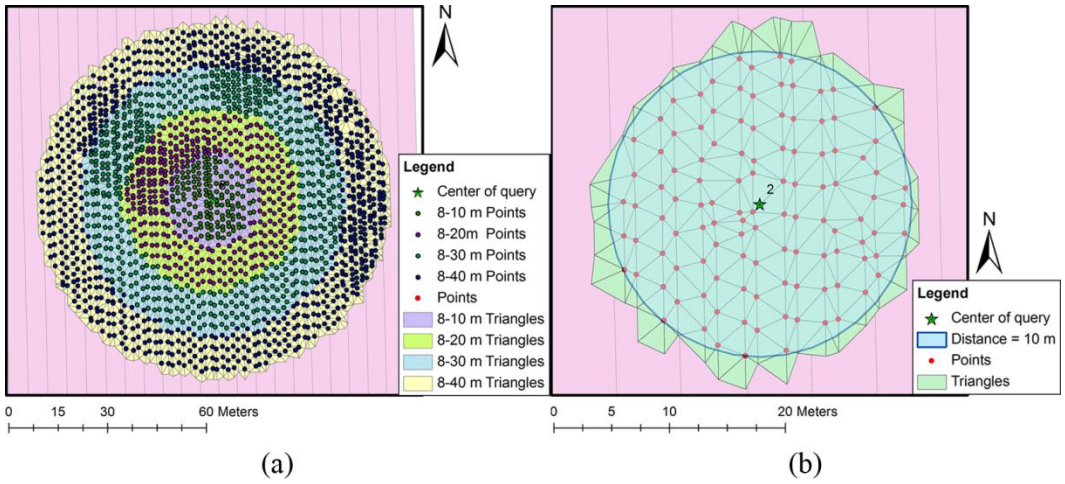


Figure 6.22. Sample results of proximity query (a) points and (b) triangles

The bounding box querying test was performed to retrieve TIN data that are within a narrow strip area. The bounding box query with narrow strips for retrieving TINs is common in visibility calculation. The query area has the width 2 m (1 m buffering around a LOS) and the varying length from 100 m to 3,000 m (i.e., 100, 200, 300, 400, 500, 1,000, 2,000, and 3,000 m). For each length, five different locations within the uploaded data area were selected. Examples of strips are shown in Figure 6.23.

The performance results are shown in Figure 6.24. Only queries for length of 100 to 500 m were completed within 30 seconds. The performance varies greatly when the length of bounding box increases due to the additional points and triangles that need to be retrieved. Similar to the proximity query, it was found that the first step of TIN query, point retrieval, varied greatly while the second step of TIN query, triangle retrieval, was relatively fast which took about 1, 2, 3, 4, and 5 seconds for the query length from 100 to 500 m, respectively.

Since the default resolution of the GeoModel's bounding box query was initially set at 10-digit (the geocell size of 38.1 x 18.9 m), which is too large for the purpose of narrow strip query, in this experiment, the resolution was set at 12-digit (the geocell size of 1.22 x 2.44 m) for refining the search result. The higher search resolution set in GeoModel, in turn, requires a larger set of computed geocells to be searched. Thus, a better performance in this experiment is observed comparing to the trend found in Experiment I.

Figure 6.25 shows sample plots of bounding box query results. The point entities around the bounding box area were retrieved, which included points inside and some points outside the querying area. This is because GeoModel returned all entities of the relevant geocells without further filtering out the irrelevant entities. Excluding the points outside the box is possible through geometric calculation but this will incur additional time. In addition, although setting the resolution lower than 12-digit, such as the default resolution at 10-digit, can yield a better performance, many extra points will be retrieved due to a larger size of geocells.

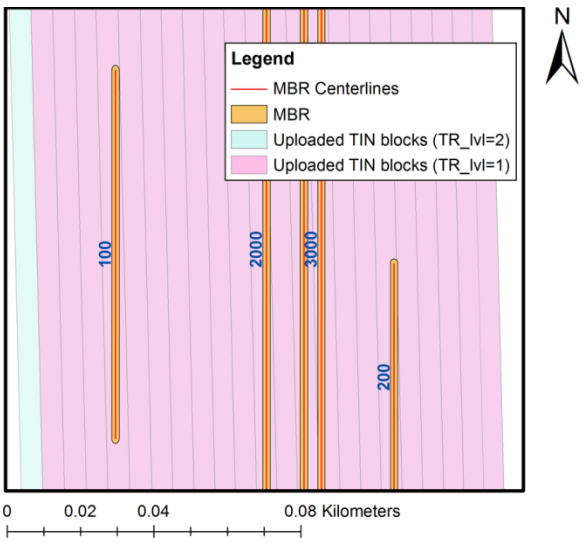


Figure 6.23. Examples of bounding boxes with varying lengths

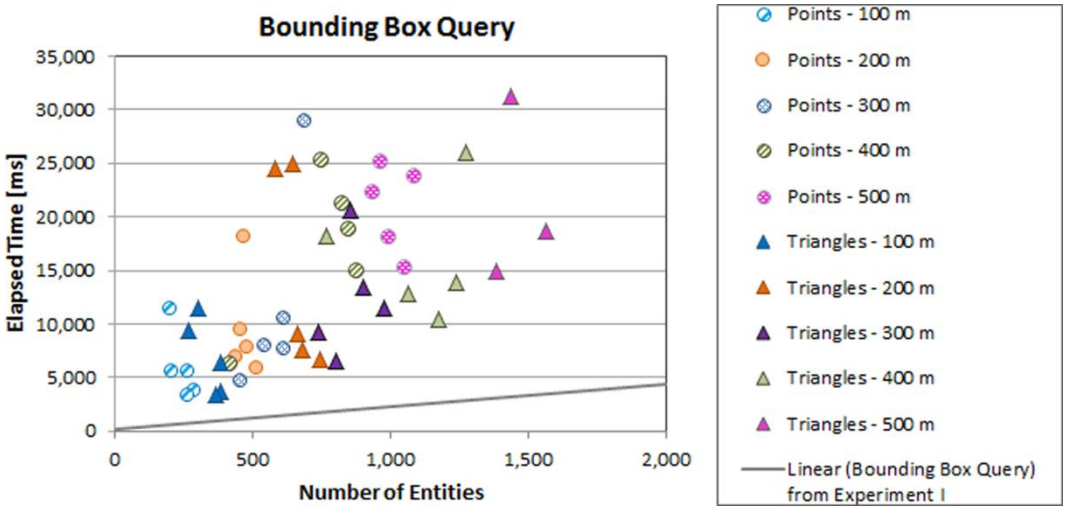


Figure 6.24. Time performance for retrieving TIN data, points and triangles, for bounding box query

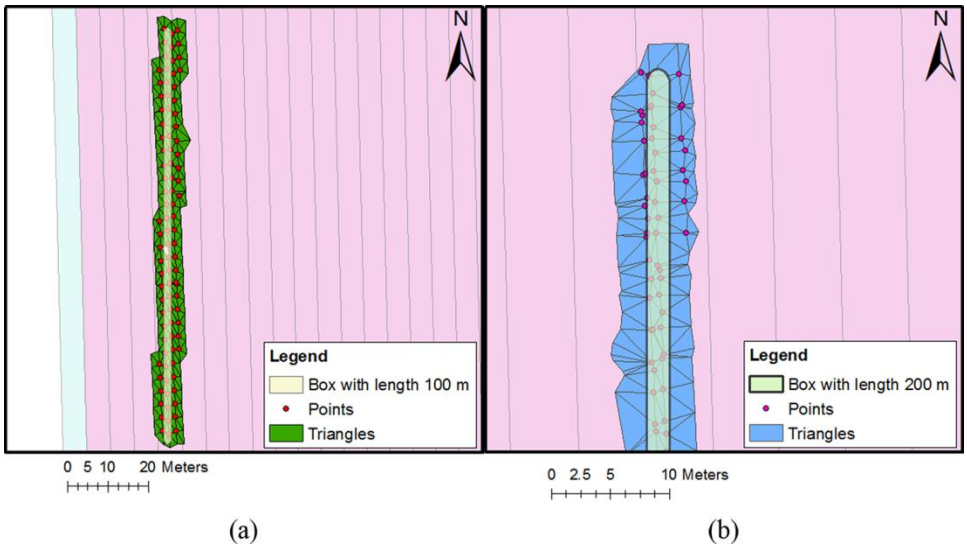


Figure 6.25. Sample results of bounding box query

In summary, our experiment for storing and retrieving TIN data using GAE and GeoModel does not entirely address the real-time performance requirement of iGNSS QoS prediction (within a few seconds). The main bottleneck is caused by spatial search for point entities, the first step of TIN query, through geocells of GeoModel. Therefore, there is a need for advanced indexing techniques and operations in GAE for efficient management and processing

of large spatial datasets, such as TINs. With such geospatial tools in place, storing and processing TIN data in GAE for real-time geoprocessing will become possible.

7.0 BENEFITS AND APPLICATIONS

7.1 INTRODUCTION

In previous chapters, we presented our methodology to predict iGNSS QoS and discussed its results for serving real-time navigation systems/services. In this chapter, we discuss its benefits, listed in Chapter 1, by analyzing two examples: (a) provision of maps with static iGNSS QoS prediction and (b) provision of optimal routing based on iGNSS QoS. Static iGNSS QoS maps help promoting awareness of QoS levels and routing based on iGNSS QoS provides alternative routes that maximize QoS along the routes. The chapter is ended by describing the usefulness of iGNSS QoS prediction to navigation systems/services as well as other applications.

7.2 STATIC IGNSS QOS MAPS

As stated in Chapter 3, there are three approaches to perform prediction: static, dynamic, and hybrid. Static iGNSS QoS prediction provides iGNSS QoS at a given time in a particular area or on particular route segments; this is particularly suitable for Web Map Services (WMSs). Dynamic prediction provides iGNSS QoS prediction on a route based on user's current location (or at specified time) and user's movement; this is particularly suitable for routing in navigation systems/services. Hybrid iGNSS QoS prediction is a tradeoff between static and dynamic predictions. Because iGNSS QoS prediction involves data- and compute-intensive tasks, as mentioned in Chapter 6, performing dynamic prediction at all locations and times may not be practical.

This section demonstrates the benefits of static iGNSS QoS prediction. Maps of geographic areas where iGNSS QoS on roads and sidewalks can be highlighted at desired times

will facilitate a new means for planning trips. Today's widely-used tools, such as WMSs, and navigation systems/services provide routes with common route choices (e.g., shortest or fastest), but there is no knowledge about GNSS positioning quality on these routes. To better understand the impacts of spatial and temporal variations on static iGNSS QoS prediction, an experiment was conducted. The iGNSS QoS parameters predicted on road and sidewalk segments at various times (static prediction) are analyzed.

7.2.1 Experiment

An experiment was conducted to simulate tracking-based iGNSS QoS parameters, i.e., $\widetilde{Av}$, $\widetilde{Ac}$, *Co*, and *Re*, on route segments presented on static maps and evaluate the results predicted at different times. Segments in different areas are used to analyze the spatial impact on iGNSS QoS prediction. To evaluate temporal impact, iGNSS QoS results were predicted: (a) at different times within a day and (b) on different dates at the same time. In addition, we investigated iGNSS QoS prediction on two networks, i.e., roads and sidewalks, to observe differences on the results for different modes of travel including driving, walking, riding bike or wheelchair.

Figure 7.1 shows the study area for this experiment, which contains road and sidewalk networks. The road network is provided by NAVTEQ (v.3 released on October 1, 2008), containing 7,325 road segments with an average segment length of 93.05 m. The sidewalk network, covering the University of Pittsburgh's main campus, containing 725 sidewalk segments with an average segment length of 43.56 m

The parameters for this experiment are set as follows. For GNSS simulation, only GPS constellation was considered using the actual broadcast ephemerides data[14]. While a combined satellite constellations is possible, the combination may introduce other factors and uncertainties that could impact the results of iGNSS QoS prediction. Other parameters for iGNSS QoS prediction suitable for different modes of travel are given in Table 7.1, as suggested by the empirical results in Chapter 5.

[14] http://igscb.jpl.nasa.gov

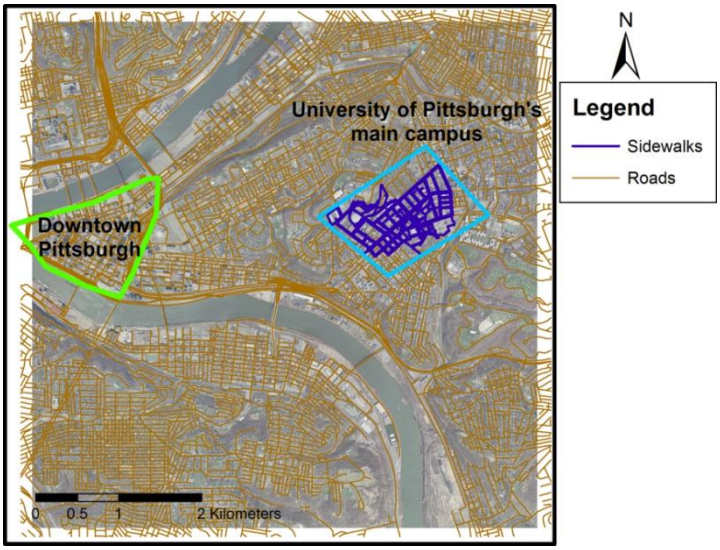


Figure 7.1. Roads and sidewalks in the study area

Table 7.1. Configuration parameters for iGNSS QoS prediction

Simulation Settings	Roads	Sidewalks
Sampling Distance (m)	60	30
Minimum required accuracy[1] ($\sigma_{threshold}$)	0.4	0.6
Granularities of TINs used in point-based prediction	2-zone LOS calculation	3-zone LOS calculation

[1] the value range from 0 to 1

iGNSS QoS variations, due to different segment locations, were observed through creation of maps and statistical results for each of the four tracking-based parameters. The statistical differences of each predicted parameter caused by temporal variation on each segment were tested using Friedman's two-way ANOVA of ranks, due to the non-normal distribution of the data.

For time variation testing, two analyses were performed. One was to predict iGNSS QoS of the entire network on the same day at four different times. The results were simulated on 5/15/2011 at 00:00, 6:00, 12:00, and 18:00 UTC. Another was to predict iGNSS QoS on selected segments every four minutes (to reflect satellite position change by one degree) for a day

(5/15/2011 from 00:00 to 23:59 UTC). A road segment and two sidewalks along the segment under each of the three environmental conditions, i.e., open sky, moderately blocked (referred as moderate), and blocked, were chosen.

For date variation testing, iGNSS QoS prediction was simulated on four different dates at the same time: 3/20/2010 (Spring), 6/21/2010 (Summer), 9/23/2010 (Fall), and 12/22/2010 (Winter) at 18:00 UTC. The experiment was controlled by using the same 3D data representing objects on the Earth's surface throughout seasons.

7.2.1.1 Time-Variant Results and Analysis

Examples of maps with iGNSS QoS prediction for road and sidewalk networks are shown in Figure 7.2 and Figure 7.3, respectively. For illustrative purposes, the value of each QoS parameter, referred to as Q, is classified into five groups: Excellent ($0.75 < Q \leq 1.0$), Good ($0.5 < Q \leq 0.75$), Fair ($0.25 < Q \leq 0.5$), Poor ($0 \leq Q \leq 0.25$), and N/A. Table 7.2 summarizes the percentage of road and sidewalk segments in each group.

Overall, a majority of roads (> 90%) in the study area have excellent QoS for all parameters at all times. Most roads with poor QoS are within the downtown area where tall buildings and urban canyons have a high impact on reception of GPS signals. On the other hand, on sidewalks, even with 90% GPS position availability, only less than 50% of sidewalks have excellent QoS. The QoS on accuracy, continuity, and reliability vary greatly on sidewalks, especially those nearby high-rise buildings located on both sides of the street. This confirms the fact that surrounding environments affect iGNSS QoS on sidewalks more frequently and more significantly than roads do.

Friedman tests revealed that the prediction results of the four QoS parameters differ significantly across the four testing times for both roads and sidewalks ($p < 0.05$). There are at least two factors contributing to QoS variations. One is satellite positions and another is surrounding environments. Under open sky condition, both factors insignificantly impact iGNSS QoS on road segments. Table 7.3 shows high percentages of road segments with no QoS variation at different times. Only a small number of road segments surrounded by tall buildings show significant iGNSS QoS variations. For most sidewalks, on the other hand, the QoS parameters varied at different times, except for the availability parameter, evidenced by high percentages of segments with different QoS in Table 7.3. The local environments surrounding

sidewalks mask satellite view at higher elevation degrees comparing to satellite view on roads. This makes the degradation of iGNSS QoS on sidewalks even more pronounced. In short, iGNSS QoS varies at different times on road segments surrounded by tall buildings and on sidewalks surrounded by objects such as buildings, trees, and cross bridges.

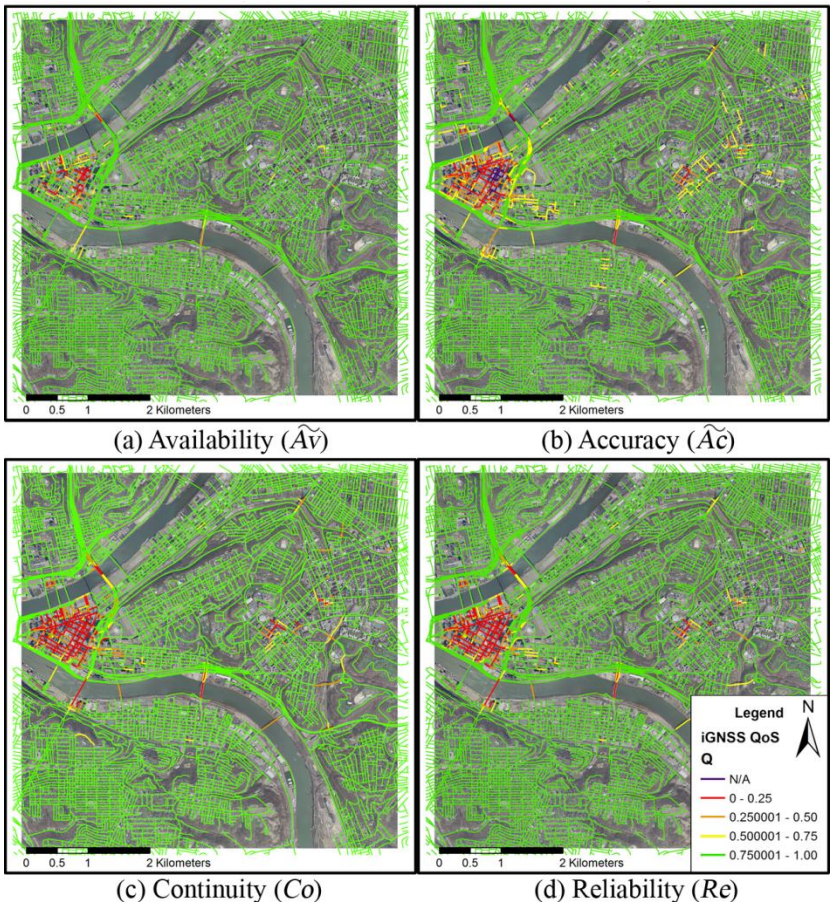


Figure 7.2. iGNSS QoS for road predicted for 5/15/2011 at 00:00:00 UTC

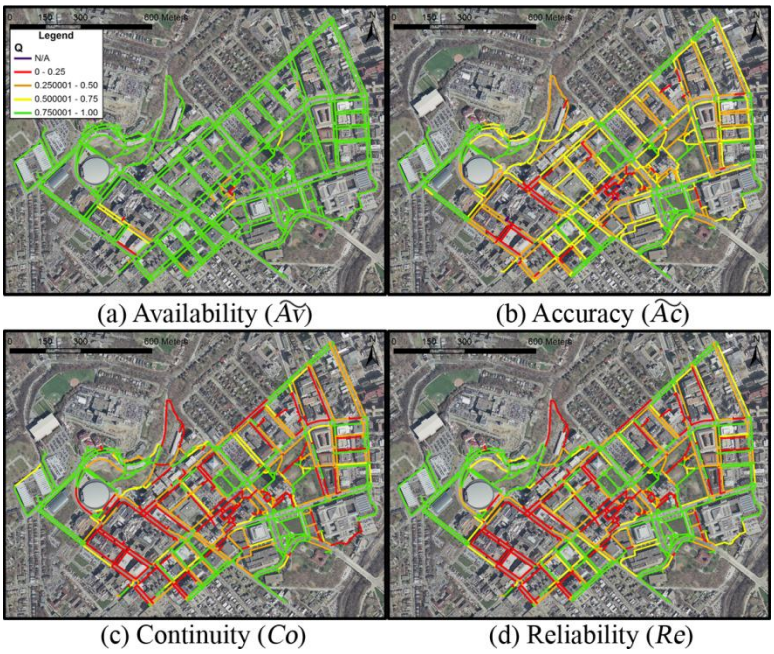


Figure 7.3. iGNSS QoS for sidewalks predicted for 5/15/2011 at 00:00:00 UTC

Table 7.2. Percentages of segments with iGNSS QoS classified into five groups predicted at different times within a day

Quality levels	**Time**	**Roads** (N = 7,325)				**Sidewalks** (N = 725)			
		$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re
Excellent ($0.75 < Q \leq 1$)	0:00	98.3%	92.6%	94.9%	95.0%	96.8%	36.0%	45.1%	46.5%
	6:00	97.1%	92.1%	93.6%	93.8%	92.3%	29.4%	38.6%	39.6%
	12:00	97.8%	92.8%	94.9%	95.1%	93.4%	47.4%	57.2%	59.3%
	18:00	96.8%	91.5%	93.2%	93.6%	90.8%	24.8%	36.1%	36.8%
Good ($0.5 < Q \leq 0.75$)	0:00	0.4%	3.3%	0.7%	0.8%	0.8%	32.7%	5.7%	6.9%
	6:00	0.5%	3.1%	0.9%	0.9%	2.2%	33.2%	5.1%	6.3%
	12:00	0.5%	3.5%	0.7%	0.9%	1.9%	31.7%	5.7%	6.6%
	18:00	0.6%	3.8%	1.0%	1.1%	2.5%	31.9%	4.4%	7.6%
Fair ($0.25 < Q \leq 0.5$)	0:00	0.7%	2.0%	1.7%	1.5%	1.9%	20.4%	24.1%	22.5%
	6:00	1.2%	2.5%	2.1%	1.9%	3.9%	25.7%	24.7%	23.9%
	12:00	1.1%	2.0%	2.1%	1.8%	3.4%	15.3%	19.9%	18.5%
	18:00	1.4%	2.0%	2.7%	2.3%	5.1%	23.4%	26.8%	24.4%

Quality levels	Time	Roads (N = 7,325)				Sidewalks (N = 725)			
		$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re
Poor ($0 \le Q \le 0.25$)	0:00	0.5%	1.7%	2.7%	2.6%	0.4%	10.8%	25.1%	24.1%
	6:00	1.2%	1.4%	3.4%	3.3%	1.7%	10.5%	31.6%	30.2%
	12:00	0.6%	1.3%	2.3%	2.2%	1.2%	4.6%	17.2%	15.6%
	18:00	1.1%	1.9%	3.1%	3.1%	1.7%	18.5%	32.7%	31.2%
N/A	0:00	0.0%	0.3%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%
	6:00	0.0%	0.9%	0.0%	0.0%	0.0%	1.2%	0.0%	0.0%
	12:00	0.0%	0.5%	0.0%	0.0%	0.0%	1.0%	0.0%	0.0%
	18:00	0.0%	0.8%	0.0%	0.0%	0.0%	1.4%	0.0%	0.0%

Table 7.3. Percentages of segments with the same or different QoS predicted at the four testing times on the same date

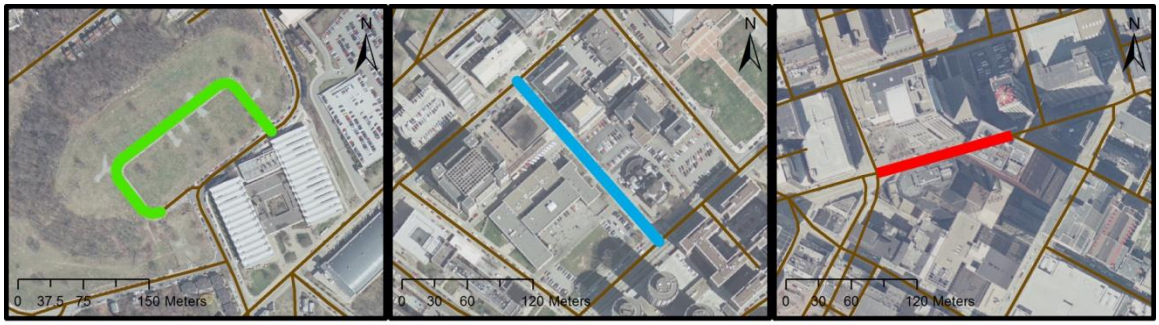
	Roads (N = 7,325)				Sidewalks (N = 725)			
	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re
Same QoS	95.5%	73.8%	92.3%	92.2%	81.9%	1.2%	27.3%	27.3%
Different QoS	4.5%	26.2%	7.7%	7.8%	18.1%	98.8%	72.7%	72.7%

For time-variant analysis of sample segments, Figure 7.4 shows the selected road segments under open-sky, moderate, and blocked conditions; they are referred to as Ro, Rm, and Rb, respectively. Both sidewalks along a street were selected under different environment settings, as shown in Figure 7.5; they are referred to as So1, So2, Sm1, Sm2, Sb1, and Sb2, respectively.

The prediction results on road segments are shown in Figure 7.6 and statistically summarized in Table 7.4. Under open sky condition, time variation has no effect on positioning QoS; the QoS parameters of the Ro segment is always excellent (= 1). Under moderate condition, time has a minor impact on position availability and causes occasional drops in continuity and reliability. The positional accuracy on the Rm segment is generally good but its value fluctuates at different times, $\widetilde{Ac} \cong 0.777$ with 0.111 deviation. Under blocked condition, although changes of iGNSS QoS on the Rb segment at different times can be observed, the

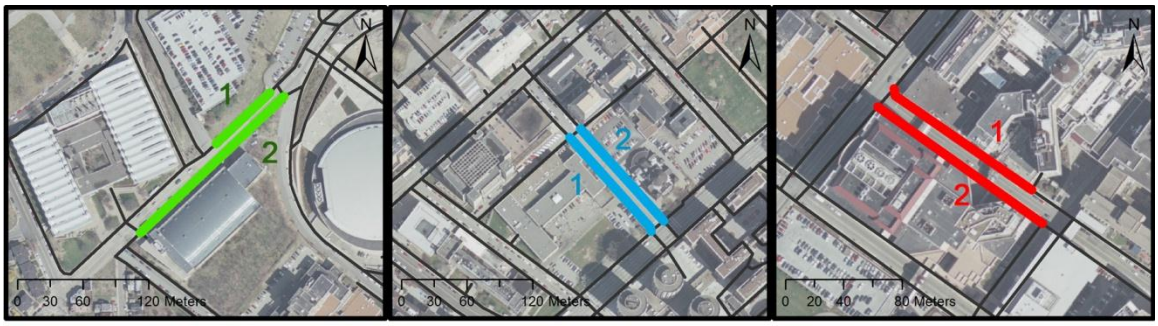
positioning quality is always deteriorated by the surrounding objects. Its positioning quality indicated by the four parameters is generally low (< 0.3), oscillating between the poor and fair group. Quality improvement from poor to good is rare. Thus, on road segments under completely blocked environments, GNSS-based applications would obtain unreliable positioning solutions most of the time.

Time and environmental factors greatly influence iGNSS QoS on sidewalks; the results are shown in Figure 7.7 and Table 7.4. Opposite sidewalks located on the same street may have different levels of positioning QoS. Segments adjacent to large buildings likely have low QoS. QoS on So1 is better than on So2, Sm2 better than Sm1, and Sb1 better than Sb2. For example, a large building on the right side of So2, see Figure 7.5(a), potentially blocks or bounces some satellite signals from time to time, which causes degradation of positional accuracy comparing to So1, see Figure 7.7. Comparing iGNSS QoS on a road (Rm) and on sidewalks (Sm1 and Sm2) located on the same street, see Figures 7.4(b) and 7.5(b), QoS are relatively lower on the sidewalks than on the road under obscured environment settings. The fluctuations can be clearly observed on continuity and reliability of these sidewalk segments (Sm1 and Sm2) with deviation about 0.3 based on the [0,1] scale. Thus, this result confirms the statement mentioned earlier that time affects greater iGNSS QoS variations on sidewalks than on roads.



(a) Open-sky (Ro) (b) Moderate (Rm) (c) Blocked (Rb)

Figure 7.4. Road segments chosen for simulating iGNSS QoS prediction for a day



(a) Open-sky (So) (b) Moderate (Sm) (c) Blocked (Sb)

Figure 7.5. Sidewalk segments chosen for simulating iGNSS QoS prediction for a day

Table 7.4. Statistics of iGNSS QoS on sample segments predicted every four minutes within a day

	Segment Name	**Length (m)**	**Mean**				**Deviation**			
			$\widetilde{Av}$	$\widetilde{Ac}$	*Co*	*Re*	$\widetilde{Av}$	$\widetilde{Ac}$	*Co*	*Re*
Roads	Ro	317.7	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000
	Rm	196.2	0.998	0.777	0.951	0.958	0.018	0.111	0.118	0.092
	Rb	127.9	0.239	0.258	0.050	0.050	0.204	0.200	0.105	0.105
Sidewalks	So1	68.1	1.000	0.986	1.000	1.000	0.000	0.034	0.000	0.000
	So2	181.5	1.000	0.881	0.935	0.970	0.000	0.083	0.170	0.082
	Sm1	114.4	0.976	0.528	0.402	0.469	0.070	0.178	0.300	0.309
	Sm2	113.8	0.985	0.601	0.520	0.588	0.057	0.170	0.306	0.294
	Sb1	118.9	0.713	0.334	0.122	0.136	0.226	0.150	0.144	0.172
	Sb2	138.8	0.437	0.287	0.036	0.039	0.204	0.131	0.060	0.067

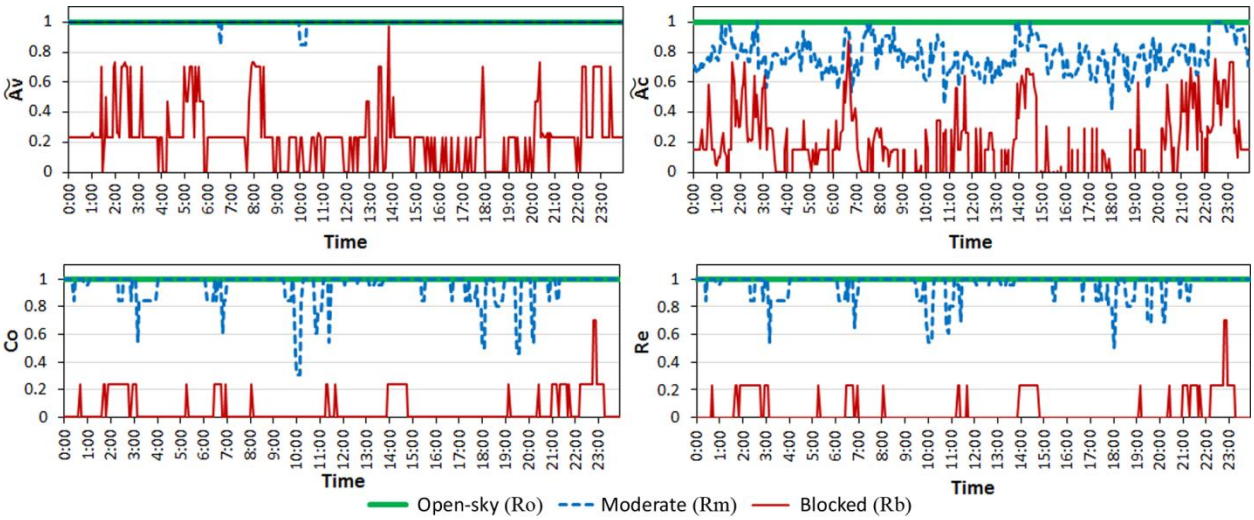


Figure 7.6. iGNSS QoS predicted every four minutes for a day on selected road segments

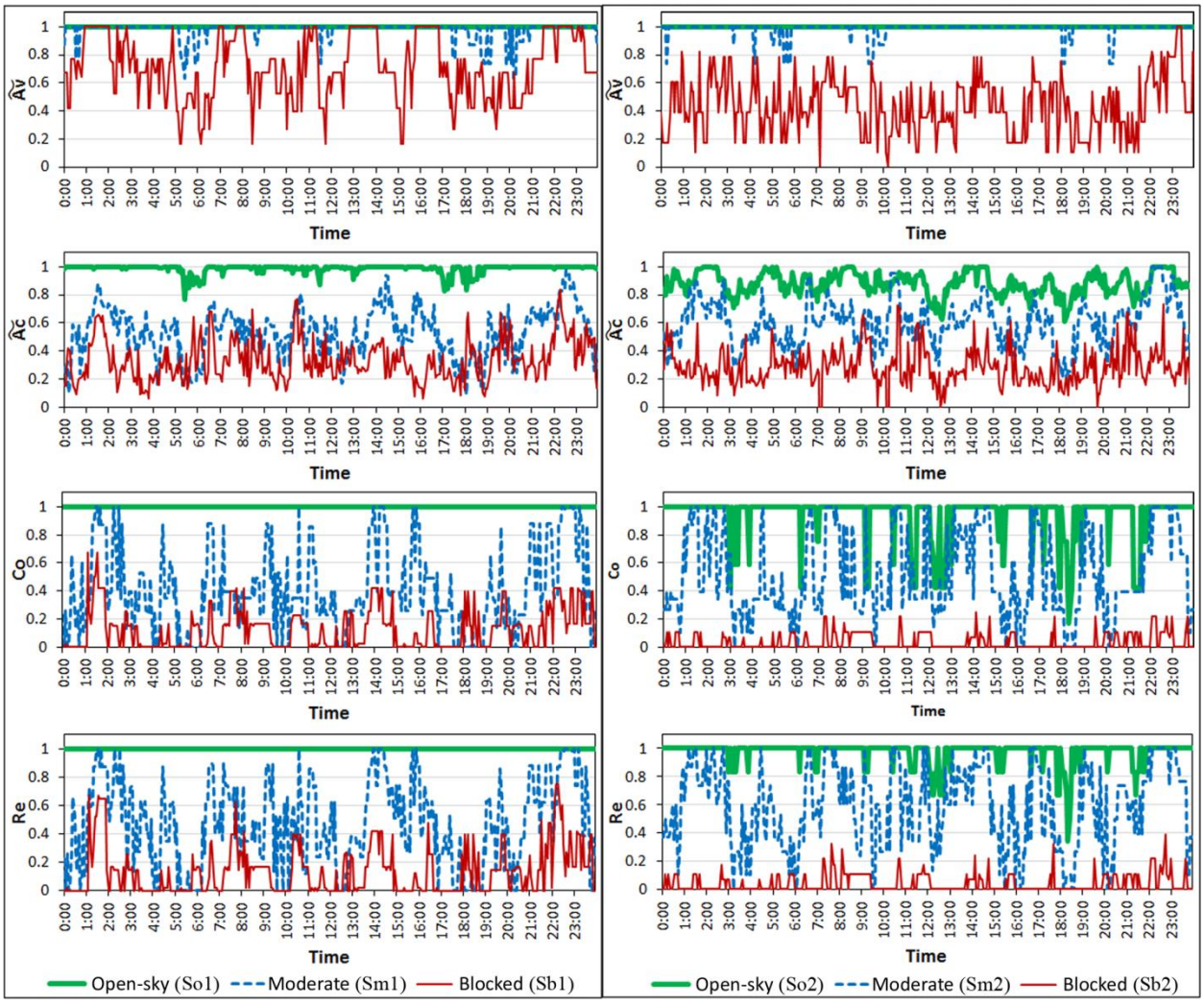


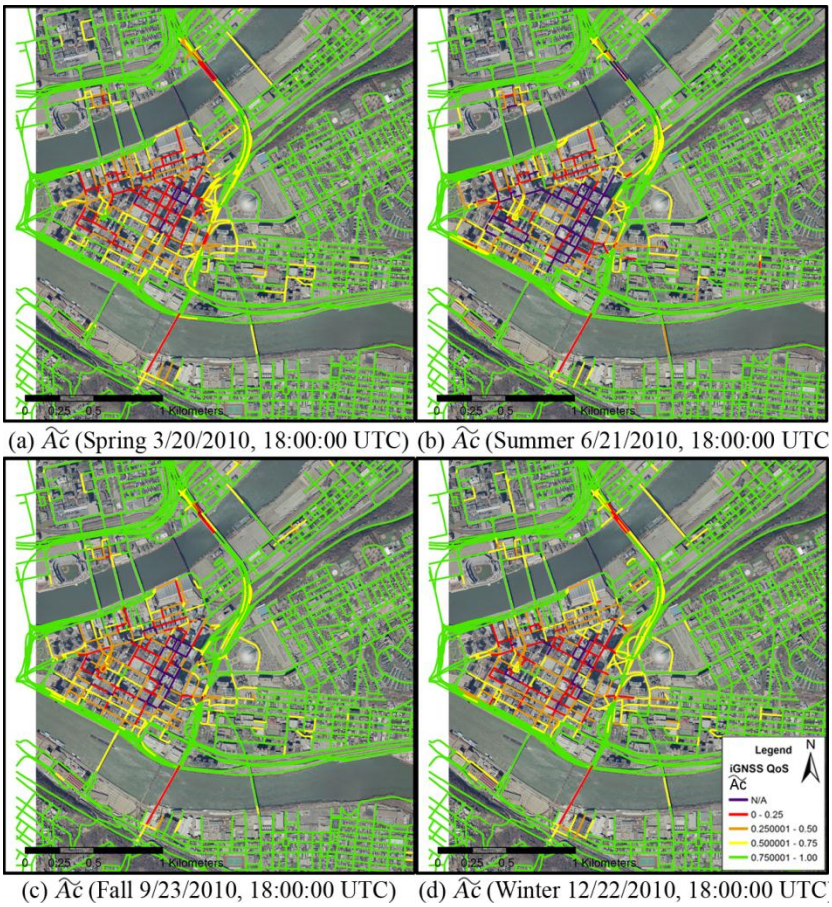
Figure 7.7. iGNSS QoS predicted every four minutes for a day on selected sidewalk segments

7.2.1.2 Date-Variant Results and Analysis

Another way to analyze iGNSS QoS variations temporally is to fix the time but change the dates. As GPS has a ground track repeat period about one sidereal day, it is skeptical whether static prediction of a certain time of the day can be used repeatedly over a time period (e.g., weeks, months).

Figures 7.8 and 7.9 show examples of maps with accuracy predicted on different dates at the same time for roads (only downtown area) and for sidewalks. The percentages of segments at each quality level are summarized in Table 7.5. Again, a majority of roads (> 90%) have excellent QoS whereas about a half of the sidewalk segments have degraded positional accuracy, continuity, and reliability.

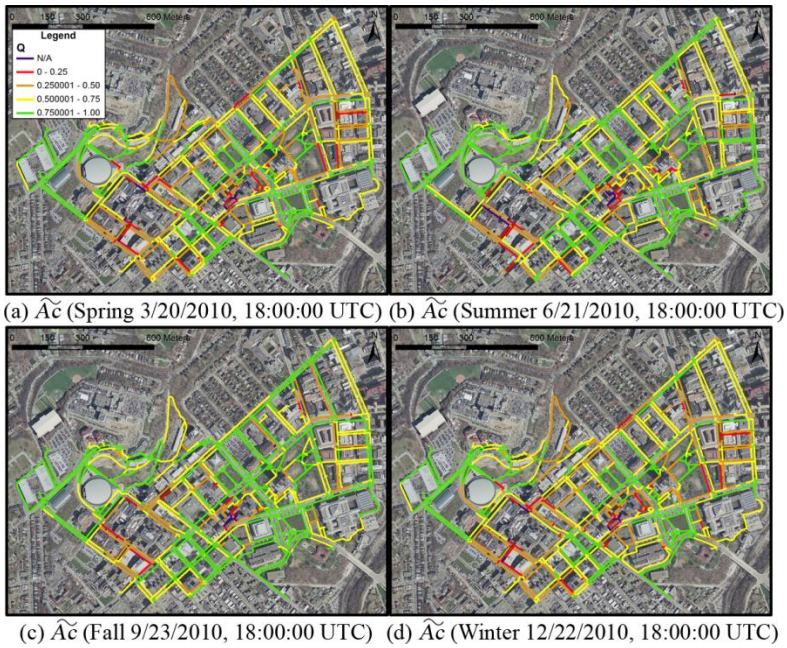
Comparing the QoS of each segment predicted on the four different dates, Friedman tests showed significant differences on the four QoS parameters for both roads and sidewalks ($p < 0.05$). The percentages of the segments with same and different QoS for this analysis are summarized in Table 7.6, which are comparable to results shown in Table 7.3. The main factor influencing differences in QoS is different sets of satellites that are available to use in the study area on different dates. It is known from the GPS theory that the satellite orbit repeats every 11 hours 57.96 minutes which causes the ground track repeat period of satellites about one sidereal day (Hofmann-Wellenhof et al., 2008). However, each GPS satellite appears at the same position about four minutes earlier each day when observed at a given location and time. The three-month span in this example caused major changes in satellite positions when observed at a fixed location and time on different dates. In addition, in practice satellites are sometimes marked as unhealthy or under maintenance. This causes different sets of available satellites to be observed which can greatly influence changes of QoS, especially in obstructed environments, due to different geometrical arrangements. Based on this experiment, 11, 7, 8 and 8 GPS satellites are available in Spring, Summer, Fall, and Winter, respectively. The PRNs of each satellite set are as follows: Spring 3,6,9,14,15,18,21,22,24,26,27; Summer 14,16,20,23,29,31,32; Fall 7,8,11,17,19,24,26,28; and Winter 2,4,5,10,12,13,25,29. Overall, the highest number of available satellites in spring contributes to the highest number of segments with excellent QoS while the lowest ones can be observed in summer due to more satellites marked as unhealthy (see Table 7.5).



(a) $\widehat{Ac}$ (Spring 3/20/2010, 18:00:00 UTC) (b) $\widehat{Ac}$ (Summer 6/21/2010, 18:00:00 UTC

(c) $\widehat{Ac}$ (Fall 9/23/2010, 18:00:00 UTC) (d) $\widehat{Ac}$ (Winter 12/22/2010, 18:00:00 UTC

Figure 7.8. Positional accuracy on road segments predicted on different dates in the downtown area of Pittsburgh



(a) $\widehat{Ac}$ (Spring 3/20/2010, 18:00:00 UTC) (b) $\widehat{Ac}$ (Summer 6/21/2010, 18:00:00 UTC)

(c) $\widehat{Ac}$ (Fall 9/23/2010, 18:00:00 UTC) (d) $\widehat{Ac}$ (Winter 12/22/2010, 18:00:00 UTC)

Figure 7.9. Positional accuracy on sidewalk segments predicted on different dates

Table 7.5. Percentages of segments with iGNSS QoS classified into five groups predicted on different dates at the same time

Quality levels	Time	**Roads** (N = 7,325)				**Sidewalks** (N = 725)			
		$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re
Excellent ($0.75 < Q \leq 1$)	Spring	99.4%	93.1%	95.8%	96.0%	99.9%	42.9%	63.7%	65.4%
	Summer	96.9%	93.3%	94.3%	94.6%	91.3%	49.2%	55.9%	57.7%
	Fall	98.0%	93.5%	95.6%	95.8%	95.3%	54.8%	60.6%	62.5%
	Winter	98.3%	92.9%	95.6%	95.8%	90.5%	36.7%	39.7%	40.6%
Good ($0.5 < Q \leq 0.75$)	Spring	0.2%	3.8%	0.7%	0.8%	0.1%	43.7%	4.8%	6.8%
	Summer	0.7%	3.0%	1.0%	1.1%	2.5%	32.0%	4.3%	6.8%
	Fall	0.4%	3.4%	0.7%	0.9%	1.5%	30.6%	5.5%	8.3%
	Winter	0.5%	3.9%	0.8%	0.8%	2.9%	34.1%	4.1%	7.4%
Fair ($0.25 < Q \leq 0.5$)	Spring	0.4%	2.0%	1.9%	1.7%	0.0%	10.2%	19.9%	18.3%
	Summer	1.3%	1.8%	2.4%	2.1%	4.8%	13.4%	23.4%	20.6%
	Fall	0.9%	1.6%	1.7%	1.4%	2.5%	11.2%	22.6%	18.8%
	Winter	0.7%	1.8%	1.8%	1.7%	5.1%	20.4%	26.3%	24.1%
Poor ($0 \leq Q \leq 0.25$)	Spring	0.1%	1.1%	1.6%	1.5%	0.0%	3.2%	11.6%	9.5%
	Summer	1.2%	1.1%	2.3%	2.2%	1.4%	4.4%	16.4%	15.0%
	Fall	0.6%	1.1%	1.9%	1.9%	0.7%	3.0%	11.3%	10.5%
	Winter	0.4%	1.2%	1.8%	1.7%	1.5%	7.6%	29.8%	27.9%
N/A	Spring	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Summer	0.0%	0.9%	0.0%	0.0%	0.0%	1.0%	0.0%	0.0%
	Fall	0.0%	0.5%	0.0%	0.0%	0.0%	0.4%	0.0%	0.0%
	Winter	0.0%	0.3%	0.0%	0.0%	0.0%	1.2%	0.0%	0.0%

Table 7.6. Percentages of segments with the same or different QoS predicted on four different dates at the same time

	Roads (N = 7,325)				**Sidewalks** (N = 725)			
	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re	$\widetilde{Av}$	$\widetilde{Ac}$	Co	Re
Same QoS	95.6%	74.9%	92.9%	92.7%	81.1%	2.5%	29.8%	29.2%
Different QoS	4.4%	25.1%	7.1%	7.3%	18.9%	97.5%	70.2%	70.8%

7.3 OPTIMAL-ROUTING BASED ON GNSS QOS

Typical routing criteria available in current WMSs and navigation systems/services are shortest and fastest. Routes generated based on such routing criteria may contain road segments with low GNSS QoS causing degradation of navigation performance. Navigation performance degradation in a navigation system/service may lead to: misidentification of the correct road segment on which the user is travelling; frequent re-routing on the same road segments; and confusions in providing instructions to the users. Provision of optimal routes based on iGNSS QoS prediction can provide routes on which navigation performance is not challenged.

Generally, there are two ways to provide optimal routes based on iGNSS QoS, as follows:

Route planning: providing route options prior to trips. Given an estimated travel date and time, e.g., hours, days, weeks, months in advance, static iGNSS QoS on route segments can be predicted. Static iGNSS QoS prediction can benefit route planning in two ways. First, it allows WMS users to realize iGNSS QoS on computed routes at a given time. Second, it gives an opportunity to WMSs for offering new routing criteria based on GNSS QoS. iGNSS QoS prediction in route planning gives flexibility for users to choose and review anticipated GNSS quality and navigation performance of a route before they travel. This will be particularly useful for planning trips in unfamiliar areas or when finding routes with high GNSS positioning quality is important.

Routing: providing routes in navigation systems/services from user's current location to a destination. With provision of dynamic iGNSS QoS prediction, reflecting most likely QoS, on travelling route, navigation systems/services can benefit in two ways: (a) re-routing on immediate segments with poor QoS and (b) being aware of unavoidable segments with poor QoS. Navigation systems/services can provide alternative routes with higher GNSS QoS that can fulfill the application's positioning requirements. Provision of iGNSS QoS on traveling routes can alert navigation systems/services about possible positioning uncertainty. Navigation systems/services can use this information to make adjustments in map matching, prepare for GNSS augmenting sensor(s) if applicable, or plan for power optimization.

To demonstrate provision of routes based on GNSS QoS, this section presents a new approach to calculate optimal routes based on four GNSS QoS criteria: maximum availability,

maximum accuracy, maximum continuity, and maximum reliability. Through two experiments, simulation and field test, it is shown that GNSS QoS-based routes are alternative routes with higher QoS computing to shortest routes.

7.3.1 Routing

Routing is an optimization problem that searches a path through a digital map from a starting point to a destination. The optimal-route problem is modeled as the minimum-cost path on a weighted directed graph. A weighted graph, *G*, consists of:

V: $\{v_1, v_2, \dots, v_N\}$ a set of vertices (or nodes) representing decision nodes including intersections or highway entry and exit points.

E: $\{e_1, e_2, \dots, e_M\}$ a set of edges (or arcs) where each edge is associated with a pair of vertices, $e = (v_i, v_j)$, representing a segment of the road (or sidewalk) between two adjacent decision nodes. In a directed graph, the order of vertices designates the traversal direction on an edge. A weight or cost, *w*, is assigned to each edge based on its attribute(s).

Given *G* = (*V, E*), an optimal route between two vertices, a source (*s*) and a destination (*d)*, is the path in *G* from *s* to *d* with the minimum cost.

A variety of weight functions ranging from simple to complex have been developed to address various user's preferences. Simple functions use a segment attribute as a weight such as distance and travel time. Complex functions combine several segment attributes to calculate a weight such as a weighted sum function for personalized driving routes in Rogers et al. (1999) and an Analytic Hierarchy Process (AHP) model for personalized wheelchair riding routes in Kasemsuppakorn and Karimi (2009b).

Several optimization algorithms such as Dijkstra's algorithm (1959) and Floyd-Warshall's algorithm (Pemmaraju and Skiena, 2003) have been used for computing optimal routes. Generally, routing algorithms are either exact or heuristic. Exact routing algorithms can guarantee best solutions (routes) but may have slow performance when searching routes in large networks (usually a large number of nodes or intersections). Heuristic algorithms, an alternative to exact algorithms, typically reduce the solution space in order to improve time complexity; for example, A* (Hart et al., 1968) and ORCA (Karimi, 1996). One drawback with heuristic algorithms is that they do not guarantee best solutions (Johnsonbaugh and Schaefer, 2003).

7.3.2 GNSS QoS-Based Optimal Route

The iGNSS QoS parameters for predicting tracking-based iGNSS QoS on each segment can be used as new routing criteria. At least four new routing criteria are possible: maximum iGNSS availability, maximum iGNSS accuracy, maximum iGNSS continuity, and maximum iGNSS reliability. We call the computed routes based on these criteria "available route", "accurate route", "continuous route", and "reliable route", respectively. The value range for each iGNSS QoS parameter is defined between 0 and 1; 1 indicates best positioning quality as required by the application, 0 worst positioning quality. In order to utilize these parameters in existing routing algorithms that minimize path's cost, the current value range of iGNSS QoS must be mapped. For this, a weight function for mapping iGNSS QoS to a suitable weight range is defined as follows:

$$w = e^{\gamma(1-Q)} \tag{7.1}$$

where Q is the value of $\widetilde{Av}, \widetilde{Ac}$, *Re*, or *Co* on each segment and γ is an impedance factor indicating the tolerance level that a navigation application can endure low iGNSS QoS. γ is a positive number (γ > 0). The higher the impedance factor, the less tolerance to poor iGNSS QoS.

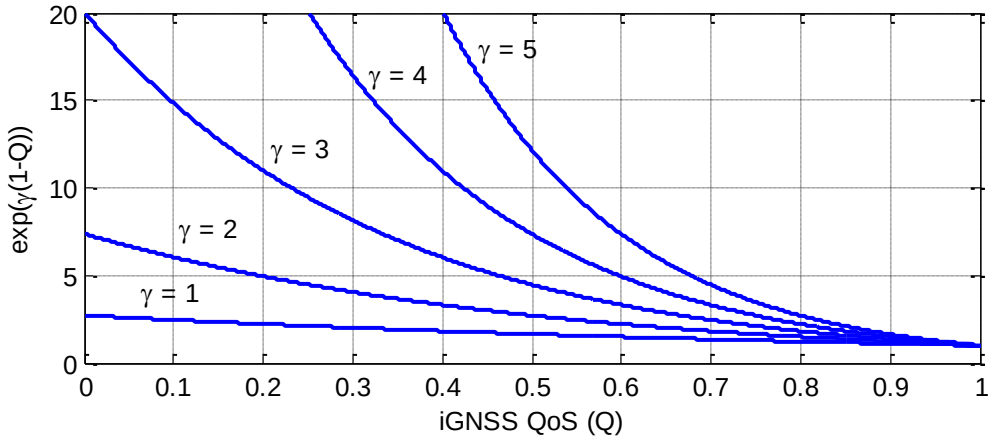


Figure 7.10. Weights based on iGNSS QoS with varying values of the impedance factor

7.3.3 Experiments

Two experiments were conducted to test GNSS QoS improvements between shortest routes and iGNSS QoS-based routes. Experiment I compared and contrasted simulated routes (i.e., shortest and four iGNSS QoS-based routes) in terms of total distance, positional availability, accuracy, continuity, and reliability. Experiment II validated GNSS QoS of sample routes (both shortest and iGNSS QoS-based routes) with the actual GPS trajectories.

Dijkstra's algorithm was implemented to find paths with minimum costs. Two modes of travel were considered in these two experiments, driving (using a road network) and walking (using a sidewalk network). The study area and networks used in this experiment are shown in Figure 7.1. The configurations of the simulation for iGNSS QoS prediction used in these experiments are summarized in Table 7.1.

In experiment I, iGNSS QoS-based routes were simulated based on static iGNSS QoS prediction (5/15/2011 at 18:00 UTC) on road and sidewalk networks. Five hundred origin and destination (O-D) pairs for driving routes were randomly selected from address points in the study area and fifty O-D pairs for walking routes were randomly selected from the points indicating the main entrance of the University of Pittsburgh's buildings. Each O-D pair was set at least 500 m apart for driving and 100 m apart for walking, both in Euclidean distance. Routes based on five criteria, shortest distance, maximum availability, maximum accuracy, maximum continuity, and maximum reliability, were generated between each O-D pair. iGNSS QoS-based routes were generated with the impedance factor γ = 1, 2, and 3. iGNSS QoS-based routes were compared against shortest routes. To compare these optimal routes, we define a ratio (η) as follows:

$$\eta_{\text{attribute}} = \frac{\sum w_c}{\sum w_{op}} \tag{7.2}$$

where $\sum w_c$ is sum of the segments' weight in the comparing route and $\sum w_{op}$ is sum of the segments' weight in the optimal route.

For example, η_{distance} is the ratio of the total distance of an iGNSS QoS-based route to the total distance of the shortest route for a given O-D pair. η_Q is the ratio of the total iGNSS QoS weight (i.e., $\widetilde{Av}$, $\widetilde{Ac}$, *Co*, or *Re*) of the shortest route to the total iGNSS QoS weight of the

iGNSS QoS-based route; the weight is computed using Equation 7.1. For example, η_{Av} is the ratio of the total availability weight of the shortest route to that of the available route. The minimum ratio, $\eta = 1$, indicates that the comparing route is as good as the optimal route.

In experiment II, two route types, shortest and reliable, were pre-planned for selected O-D pairs. For driving, three O-D pairs located in open sky, mixed environment, and high-rise buildings environment were selected. For walking, only one O-D pair was selected. Reliable routes for driving were calculated using a digital road map with static iGNSS QoS predicted on 5/15/2011 at 18:00 UTC. For walking, a digital sidewalk map with static iGNSS QoS predicted on 5/15/2011 at 00:00 UTC was used. To evaluate the routes computed based on iGNSS QoS prediction, GPS points on each route were collected within ± 1 hour of the prediction time. A GPS receiver with E-TEK EB85A with -158 dBm sensitivity and 3.3 m CEP accuracy (no DGPS aided) was used in this experiment. The receiver was set to compute a position every second. For driving, the receiver was placed on the car's dashboard and for walking it was placed on the data collector's palm at shoulder height (about 1.2 m) with the horizontal direction parallel to the ground. The differences between shortest and reliable routes using the actual quality of GPS points are statistically compared. The GPS trajectories are also visually inspected by using iGNSS QoS chunks.

7.3.3.1 Simulated Routes: Results and Analysis

The distance ratios for comparing QoS-based routes and shortest routes are presented in Figure 7.11. Overall, increasing the impedance factor of QoS-based routes results in longer distances comparing to shortest routes. With a higher impedance factor, segments with low QoS are less tolerant by navigation system/service, thus the algorithm chooses the longer path in order to minimize poor QoS (maximize good GNSS QoS). With $\gamma = 1$, see the left most column of the histograms in Figure 7.11(a) for driving and (b) for walking, the 95th percentile of each of the QoS-based routes is about 50% longer than shortest routes for both driving and walking routes. With $\gamma = 2$ and 3, the longer routes with higher η_{distance} can be observed. For $\gamma = 3$, the 95th percentile of each of the QoS-based routes is about 70% and 110% longer than shortest routes for driving and walking, respectively. Walking routes generally have higher η_{distance} than driving routes because lower GNSS QoS often occurs on sidewalks than roads, see Figures 7.2 and 7.3. iGNSS QoS-based routes with high η_{distance} (> 1.5) occur for the O-D pairs whose shortest routes

pass through poor GNSS QoS areas. Routes with low η_{distance} (< 1.5), shortest and QoS-based, are mostly located in open sky areas.

Comparing among QoS-based routes, the results showed that available routes are relatively shorter than other QoS-based routes, with lower η_{distance} (see the first row of the histograms in Figure 7.11). This is because most road and sidewalk segments in the study area have high GNSS position availability (low $\widetilde{Av}$ weight), see Figure 7.2(a) for driving and Figure 7.3(a) for walking, thus a small number of segments are included for minimizing the total $\widetilde{Av}$ weight. The distance difference can be easily observed in walking routes with γ = 3 in which the 95th percentile of available routes is about 90% times of shortest routes while the 95th percentile of each of the other QoS-based routes is about 110% times of shortest routes.

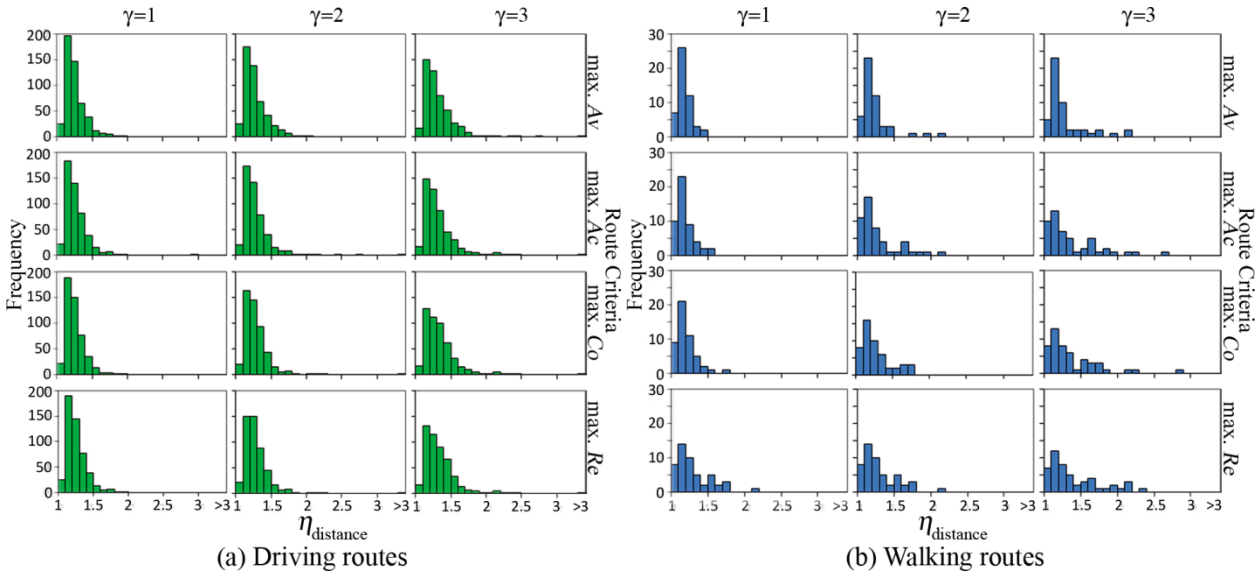


Figure 7.11. Distance ratios of iGNSS QoS-based routes and shortest routes

Figure 7.12 shows the results of GNSS QoS ratios for comparing QoS on shortest routes to QoS-based routes. Overall, as the impedance factor increases, the GNSS QoS of QoS-based routes greatly improve over that of shortest routes. With γ = 1, the 95th percentile of available, continuous, and reliable routes for driving improve GNSS QoS over that of shortest routes by 60% and, for accurate driving routes, the overall positional accuracy is improved by 90%. The

QoS improvements for walking routes when γ = 1 are 60% for available routes, 80% for accurate routes, and 100% for continuous and reliable routes at the 95th percentile. Greater improvements can be observed when γ = 3 in which 4.8% of available routes, 16.8% of accurate routes, 13.6% of continuous routes and 13.6% or reliable routes have $\eta_Q > 3$ (QoS improved by over 200%) whereas 2%, 12%, 4%, and 18%, respectively, for walking routes have $\eta_Q > 3$. Because GNSS QoS on road segments is generally good, only a small proportion of the QoS-based driving routes has significant QoS improvements. Major improvements of QoS-based routes occur on routes that avoid problematic zones such as downtown areas, comparing to substantial QoS drops on shortest routes that pass through problematic zones.

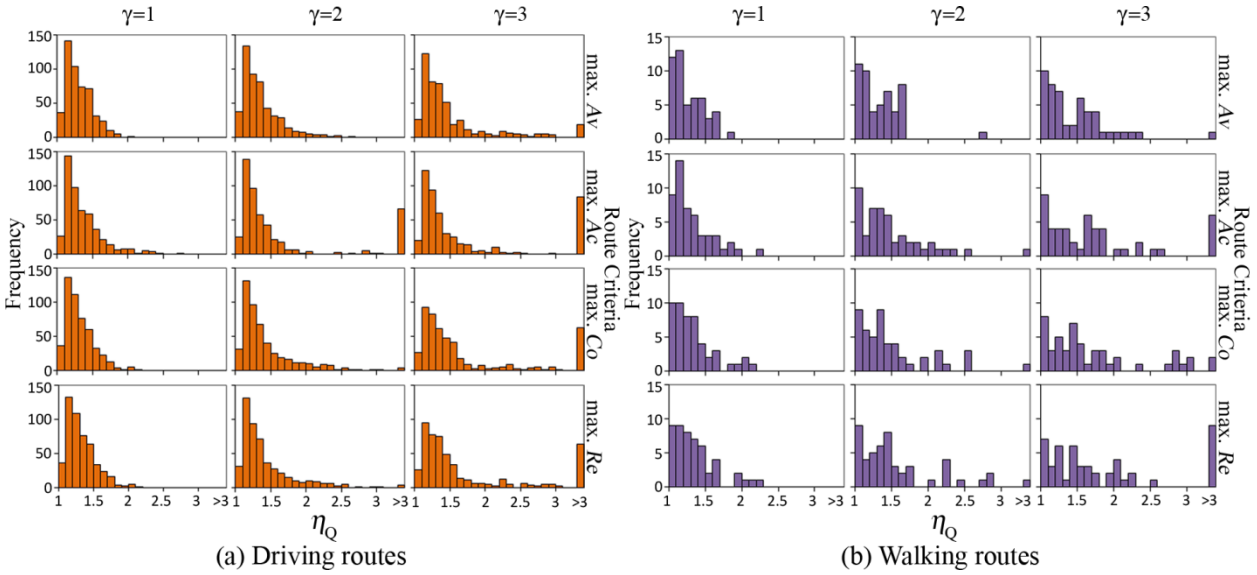


Figure 7.12. GNSS QoS ratios of shortest routes to iGNSS QoS-based routes

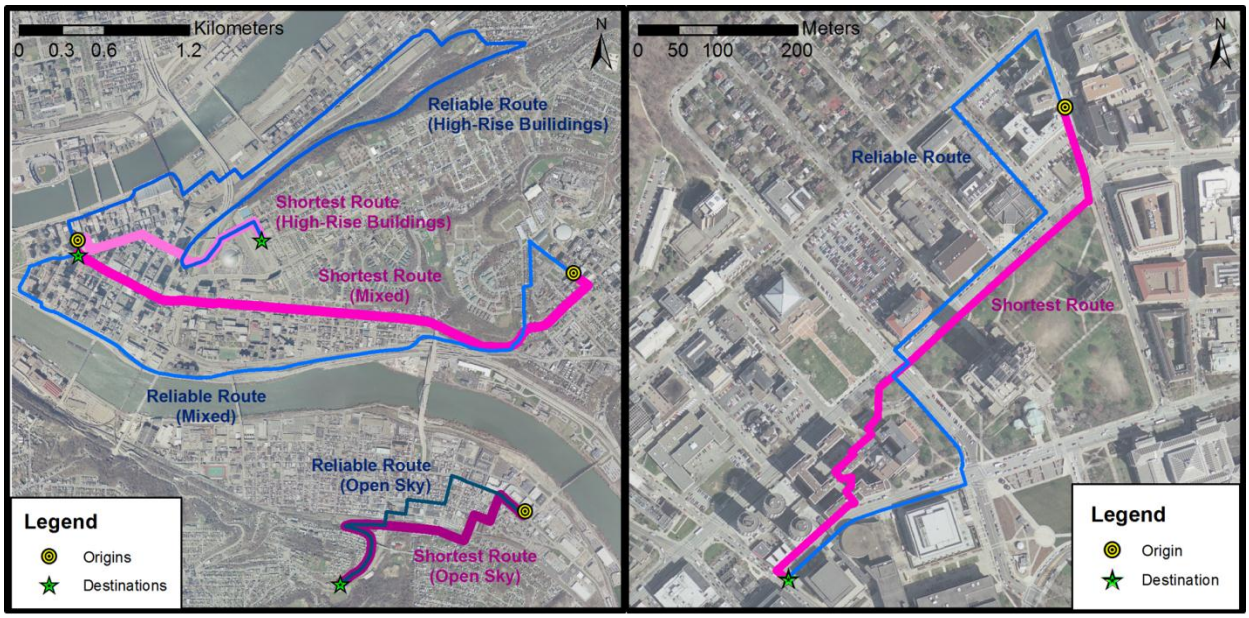
7.3.3.2 Field Test: Results and Analysis

The shortest and reliable routes generated for selected O-D pairs are shown in Figures 7.13(a) and (b) for driving and walking routes, respectively. Reliable routes were generated with γ = 3 to ensure high GNSS QoS. The total distance of each route is shown in Table 7.7. Reliable routes are less than 1.5 times of shortest routes, except for high-rise building routes. Table 7.7 also shows the date and time at the start of each GPS trajectory. In this experiment, GPS trajectories

were collected by driving in three consecutive days to control the travel time to be within ±1 hour of static GNSS QoS prediction used for calculating QoS-based routes. To reflect GNSS QoS at the time of travel, iGNSS QoS chunks for each route was predicted at the date and time of data collection. The results of QoS chunks are presented in Figures 7.14(a), (c), (e), and Figure 7.15(a).

Table 7.7. Characteristics of selected routes and trajectories

Travel Mode	Environmental Condition	Shortest Route			Maximum Reliability Route		
		Start of GPS Trajectory		Distance [km]	Start of GPS Trajectory		Distance [km]
		Date	Time		Date	Time	
Driving	Open sky	6/5/2011	17:59:40	2.21	6/5/2011	18:13:06	2.30
	Mixed	6/3/2011	17:48:16	4.04	6/3/2011	18:26:15	5.52
	High-rise buildings	6/4/2011	17:56:03	1.63	6/4/2011	18:38:36	7.96
Walking	-	6/3/2011	23:33:34	0.82	6/3/2011	23:54:58	1.16



(a) Driving routes (b) Walking routes

Figure 7.13. Routes used in testing

Table 7.8 summarizes the quality of GPS trajectories by using two metrics: GPS availability and distance errors. GPS availability of each trajectory is the percentage that a GPS point is available every second. Distance error is measured from the GPS point to the nearest point on the correct segment identified by the data collector's visual inspection.

Table 7.8. Quality of GPS trajectories

Travel Mode	Environmental Condition	Shortest Route			Maximum Reliability Route		
		GPS Availability	GPS Distance Error [m]		GPS Availability	GPS Distance Error [m]	
			Mean	Max.		Mean	Max.
Driving	Open sky	100%	2.27	6.10	100%	3.86	9.56
	Mixed	100%	7.86	43.53	99.75%	3.77	12.92
	High-rise buildings	97.94%	41.58	190.88	100%	6.73	28.12
Walking	-	100%	3.40	25.08	100%	2.20	12.07

For driving routes under open sky view, GPS trajectories of both route types are excellent, see Figure 7.14(b), which agree with the predicted QoS chunks given in Figure 7.14(a). The GPS distance errors are minimal with an average of 2.27 m and 3.86 m for shortest and reliable routes, respectively. However, the reliable route has a slightly greater distance error partly due to the use of static QoS prediction on different dates which may not reflect the actual QoS at the collection time. Figure 7.14(a) shows a chunk with good QoS ($\widetilde{Ac}$ = 0.55) highlighted in yellow on the reliable route, which shows minor drifts on the GPS trajectory.

Under mixed environmental condition, a portion of the shortest route passes through the downtown area where many high-rise buildings are located. The QoS chunks, shown in Figure 7.14(c), indicate possibilities of poor or unavailable GPS solutions in the later part of the shortest route whereas the QoS chunks of the reliable route are excellent for most parts of the route except a small chunk near the destination that has a high percentage of GPS unavailability. Figure 7.14(d) shows a close-up of the two GPS trajectories in the problematic area. It can be observed that many GPS points deviate from the shortest route with the maximum distance error

of 43.53 m. On average, the distance error of the shortest route is 7.86 m versus 3.77 m for the reliable route under mixed-environmental condition.

For the shortest driving route that passes through the high-rise buildings zone, more than half of the QoS chunks indicate poor QoS, see Figure 7.14(e). On the other hand, the reliable route has excellent QoS for most of the route except some segments near the start. However, the reliable route has longer distance due to road restrictions and poor GNSS QoS in the downtown area. As shown in Figure 7.14(f), large deviations of GPS points can be observed on the shortest route. The problem occurred at the beginning of the trip where the maximum distance error of 190.88 m and the lowest GPS availability of 97.94% are observed. The GPS trajectory of the reliable route shows accurate GPS points with the average distance error of 6.73 m; minor deviations can be observed at the beginning of the trip.

Considering walking routes, the origin is located at the middle of a blockage area which causes the reliable route to select the north path for a better GNSS QoS while the shortest route chooses the south path, as shown in Figure 7.15(a). The major GNSS QoS drops on the shortest route are expected on the pedestrian paths between tall buildings (not along the road) toward the destination. Figure 7.15(b) shows the two GPS trajectories where the shortest route has the maximum error of 25.08 m at the location predicted by the iGNSS QoS chunks. Deviations on the reliable route can also be observed due to tall buildings located nearby the sidewalks.

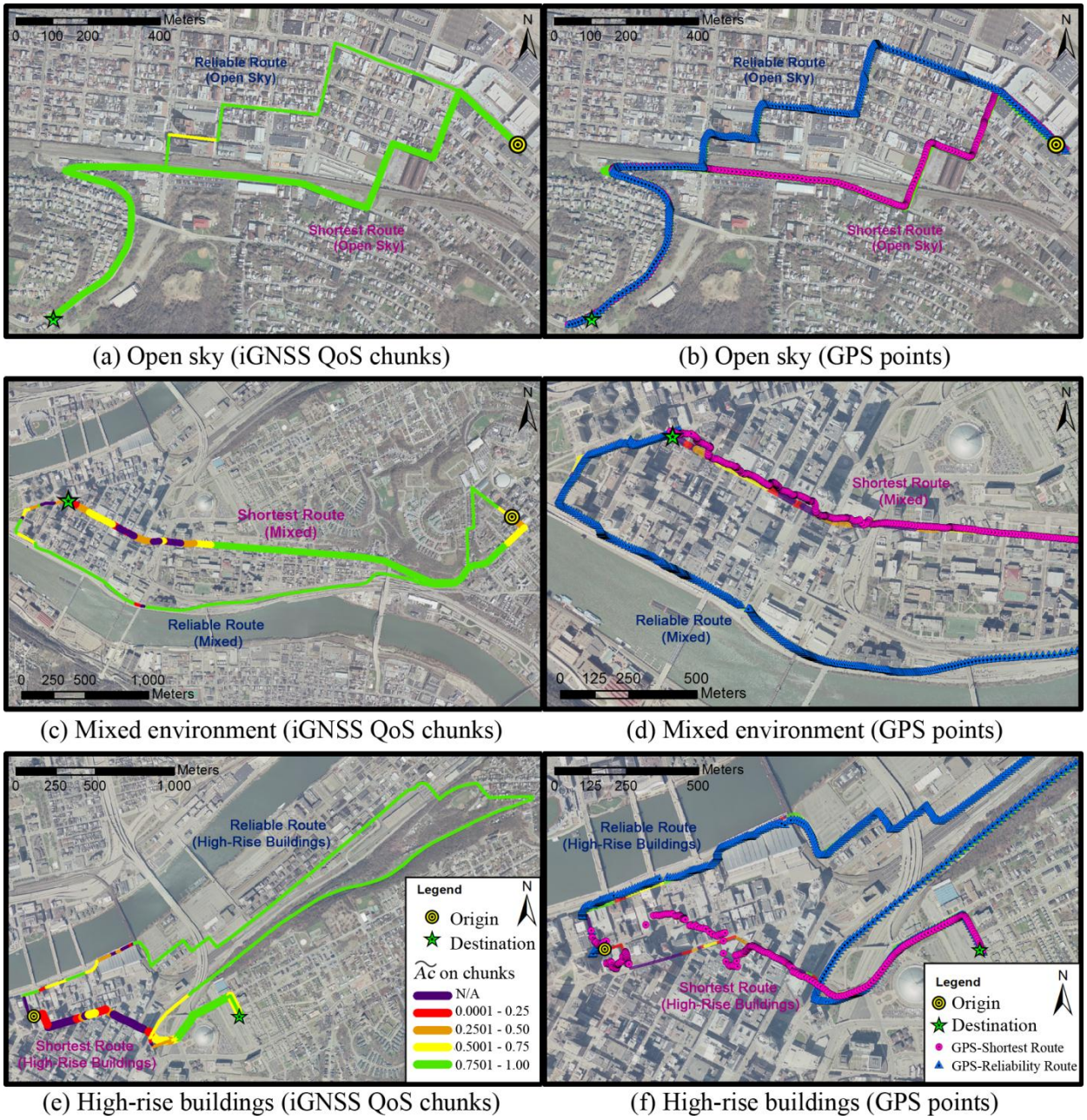
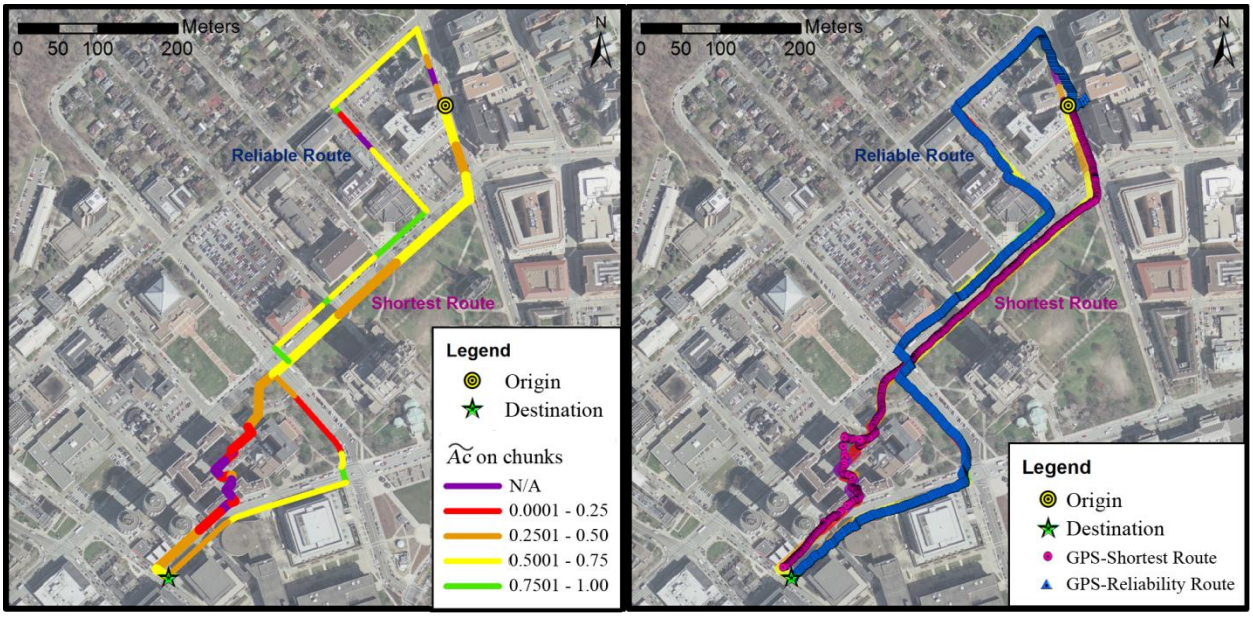


Figure 7.14. iGNSS QoS chunks and GPS trajectories for driving routes



(a) Open sky (iGNSS QoS chunks) (b) Open sky (GPS points)

Figure 7.15. iGNSS QoS chunks and GPS trajectories for walking routes

7.4 APPLICATIONS

To better understand the contributions of this research, the followings are some of the potential applications that can benefit from iGNSS QoS prediction. We first describe the benefits of iGNSS QoS prediction to navigation systems/services as they are the focus of this research. Then, other applications, Nav2Nav, data collection, and system planning and development using iGNSS hot spots are described.

7.4.1 Navigation

iGNSS QoS prediction can benefit car, pedestrian, and wheelchair navigation applications in several ways: route planning/routing, iGNSS quality awareness, GNSS augmentation planning, and energy optimization.

First, provision of maps with static iGNSS QoS prediction allows applications, including route calculation in navigation systems/services and trip planning, to plan optimal routes that

provide GNSS positioning quality ahead of time (e.g., hours, days, or weeks in advance). In addition to typical routing criteria (i.e., shortest and fastest) offered by current navigation systems/services, new routing criteria based on positional availability, accuracy, continuity, and reliability are possible. For example, one may want to plan a route with high GNSS reliability to an unfamiliar location in the downtown area; a company may plan for a route with high GNSS continuity in order to continuously tracking locations of their products. In case the user deviates from the computed route, the navigation application can instantaneously re-route the user to a new route with high iGNSS QoS.

Second, iGNSS QoS prediction allows the navigation application to be aware of iGNSS QoS along the route, especially in the unavoidable areas with poor iGNSS QoS. Thus, applications can know ahead of time the GNSS QoS of particular areas. This prior information will eliminate the possibility of surprise and confusion and allows navigation applications to have a better judgment for rejecting or accepting GNSS solutions used in subsequent navigation modules (i.e., map matching and re-routing in problematic areas).

Third, knowing iGNSS QoS in advance can help navigation applications to prepare for geo-positioning augmentation, if needed. It is anticipated that the next generation of mobile devices will be equipped with several geo-positioning sensors, such as iGNSS, Wi-Fi, camera, digital compass, and accelerometer. Not all sensors need to operate continuously and simultaneously but they will be ready to augment iGNSS when needed. For example, a wheelchair navigation application with a hybrid positioning system (GPS and vision-based positioning system) can activate the camera and prepare reference images in the areas with poor GPS and deactivate VPS whenever GPS provides high QoS.

Finally, iGNSS QoS prediction allows navigation applications to manage and optimize energy. Selecting and activating appropriate geo-positioning sensors capable of providing positions that meet the navigation application’s requirements is one way to optimize power consumption. Moreover, given that iGNSS receivers will use signals transmitted from different satellite constellations, tracking all available satellites simultaneously, potentially up to 50 from the four GNSS constellations, requires a larger amount of power compared with when a single GNSS is used. With iGNSS QoS prediction, the application can decide which combination of GNSSs will provide best positioning solutions while optimizing power consumption.

7.4.2 Nav2Nav

Nav2Nav is an ongoing research carried out in the Geoinformatics Laboratory, the School of Information Sciences at the University of Pittsburgh (Karimi et al., 2010). The concept of Nav2Nav is to allow PNDs, in a vicinity to communicate and exchange navigation information for the purpose of crash avoidance at intersections. As the crash avoidance module of the hosting car determines crash scenarios based on other vehicles' current location, the positioning quality of GNSS plays a crucial role in ensuring the true location of each vehicle.

With the use of iGNSS QoS prediction, each car can broadcast its position along with the iGNSS QoS for the perspective route segments. This will allow the crash avoidance module to make better decisions based on the known level of positioning uncertainties. For example, in Figure 7.16, the PNDs of Car 1 and Car 2 know their iGNSS QoS as they approach the same intersection. The PND of Car 1 experiences poor GNSS QoS in the communication zone, so it decides to send its current position along with a warning of upcoming poor iGNSS QoS to Car 2 before it enters the communication zone. As Car 1 enters the poor iGNSS QoS area, the crash avoidance module in Car 2 sends a warning message to its driver about the potential situation with Car 1.

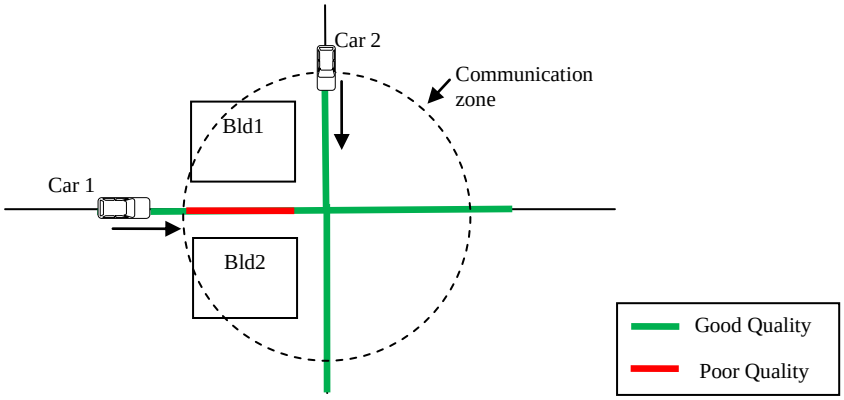


Figure 7.16. Positional quality awareness in Nav2Nav

7.4.3 Data Collection

Data collection in many applications can benefit from iGNSS QoS prediction. One example application is collaborative mapping for constructing pedestrian networks proposed by Kasemsuppakorn and Karimi (2009a). Polylines representing sidewalks are constructed from a pool of GNSS trajectories collected by volunteers using GPS-enabled mobile devices and stored through a social navigation network, called SoNavNet (Karimi et al., 2009). By providing maps with iGNSS prediction highlighted, it can help identify areas with high reliability of GNSS data before asking the collectors for trajectories; it can be used for scheduling the best time for data collection to ensure high positioning accuracy; and it can be used as a quality indicator to further filter out the collected GNSS points with low quality. As a result, only high quality GPS points will be included in the network construction process to ensure high accuracy of sidewalk segments. Another example application is road centerline mapping by using Mobile Mapping Systems (MMSs) (Toth and Grejner-Brzezinska, 2004). A MMS is a kinematic platform equipped with multiple imaging and positioning sensors including GNSS for collecting geospatial data of road features. With an approach presented in Karimi and Grejner-Brzezinska (2004), iGNSS QoS prediction can support planning optimal data collection using MMSs in order to improve mapping performance and productivity while reducing operational costs.

7.4.4 iGNSS Hot Spots

It is envisioned that iGNSS QoS prediction will facilitate the emergence of many new applications. One example is iGNSS hot spots. The idea of iGNSS hot spots is to provide maps containing a snapshot of iGNSS QoS prediction or statistics of iGNSS QoS prediction over a time period. A wide range of GNSS-based applications can benefit from these maps. One example application is GNSS-based Road User Charging (RUC) (Jensen et al., 2005, Vrhovski et al., 2004a). Determining based on GNSS points, motorists are charged for entering or driving within the charging area. To ensure the demanded levels of charging performance, Toledo-Moreo et al. (2010) suggested three approaches for complementing GNSS, aiding positioning sensors, exploiting information from enhanced maps, and providing charging integrity. In addition to these, provision of maps with iGNSS QoS prediction can be used: (a) to identify

reliable charging sites and (b) to support calculation of charging integrity to avoid incorrect charges.

8.0 SUMMARY, CONCLUSIONS, CONTRIBUTIONS AND FUTURE RESEARCH

8.1 SUMMARY

Navigation applications, among other location-based applications that rely heavily on GNSSs, are subject to positioning uncertainties. As users increasingly rely on mobile devices for navigation assistance, they expect highly reliable navigation performances, especially in complex urban areas where quality of GNSS positioning solutions are often deteriorated. To recommend navigation based on GNSS positioning quality, this research provides a methodology, called iGNSS QoS prediction, for navigation services. The premise of the methodology is provision of GNSS QoS for selected or all routes in a given area ahead of time. The methodology is not meant to mitigate errors or improve performances of iGNSS but it provides knowledge about GNSS positioning quality that allows navigation systems/services, as well as other GNSS-based applications, to prepare a suitable plan before the users reaches areas with poor iGNSS QoS.

To measure positioning quality suitable for navigation applications, six parameters are defined for iGNSS QoS: visibility, availability, accuracy, continuity, reliability, and flexibility. The methodology encompasses four modules: segment sampling, point-based iGNSS QoS prediction, tracking-based iGNSS QoS prediction, and iGNSS QoS segmentation. Given a planned route, the output of the methodology is iGNSS QoS on route segments.

The segment sampling module systematically selects points on a route (or a series of segments). Initially, the systematic sampling method with a fixed distance is applied. Point-based iGNSS QoS prediction consists of an iGNSS simulation and signal propagation models. A set of algorithms are developed for determining visibility of different LOS types using high-resolution TINs constructed from LiDAR point cloud. This module computes three iGNSS QoS parameters, i.e., visibility, availability, and accuracy, on each sample point at a specified time. To predict quality levels of positional accuracy, a fuzzy logic model was developed to combine several

factors impacting GNSS accuracy. The tracking-based iGNSS QoS prediction module is responsible for estimating average availability ($\widetilde{Av}$), average accuracy ($\widetilde{Ac}$), continuity (*Co*), and reliability (*Re*), QoS on each route segment based on point-based QoS prediction. To reduce redundancy of point-based QoS information while indicating locations where iGNSS QoS variations occur along the route, the iGNSS QoS segmentation module generates QoS chunks. A chunk is a sequence of sub-segments with similar iGNSS QoS on a route segment. A new algorithm was developed to determine the consecutive sampling points in a chunk that has a variation of iGNSS QoS within a tolerance threshold. Several experiments were conducted to evaluate the results of iGNSS QoS prediction by comparing against the actual GPS data in various environment settings and modes of travel (i.e., walking and driving).

To address time performance of iGNSS QoS prediction, two approaches were investigated. One approach is to reduce the number of iGNSS satellites included in simulating the prediction. A new algorithm, MCSSA, was developed for selecting a subset of satellites that provides near-optimal GNSS quality. Another approach is to employ distributed computing platforms for simulating iGNSS QoS prediction. Grids and clouds were explored. For the grid experiment, PittGrid was used to evaluate performance of iGNSS QoS prediction by focusing only on direct LOS calculations. For the cloud experiment, GAE was used to evaluate performance of retrieving TIN data from its distributed data storage.

Two examples of iGNSS QoS prediction benefits were shown: maps with static iGNSS QoS prediction and with dynamic iGNSS QoS-based optimal routing. Provision of maps with static iGNSS QoS prediction will facilitate a new method of making navigation decisions by providing awareness of low and high positioning quality through segments of areas and supporting route planning based on iGNSS QoS. To understand variations of iGNSS QoS prediction across space and time, the results based on both spatial and temporal factors were analyzed. Optimal routing based on iGNSS QoS can provide routes with high iGNSS QoS. Four new routing criteria based on iGNSS QoS are: maximum availability, maximum accuracy, maximum continuity, and maximum reliability. Improvements of positioning quality on iGNSS QoS-based routes are shown through simulation results and actual GPS trajectories. In addition to these examples, benefits of iGNSS QoS prediction to navigation systems/services as well as other GNSS-based applications that could take advantage from this research were described.

8.2 CONTRIBUTIONS

The main contributions of this research can be summarized as follows:

- Definition of six iGNSS QoS parameters: visibility, availability, accuracy, continuity, reliability, and flexibility, suitable for navigation systems/services (Chapter 3).
- A new methodology, iGNSS QoS prediction, which encompasses four modules where several algorithms and models are developed:
    - A fuzzy logic model for accuracy prediction (Chapter 4),
    - A set of algorithms for determining visibility of each signal path type (Chapter 4),
    - An algorithm for assigning iGNSS QoS chunks (Chapter 5).
- An efficient algorithm for iGNSS satellite selection to reduce the number of satellites in visibility calculations (Chapter 6).
- Feasibility studies of using grid-based and cloud-based simulations to address real-time performance of iGNSS QoS prediction (Chapter 6).
- Provision of maps with iGNSS QoS prediction (Chapter 7).
- Optimal iGNSS QoS-based routing in WMSs and navigation systems/services (Chapter 7).

8.3 CONCLUSIONS

iGNSS QoS prediction presented in this dissertation has the potential to provide navigation recommendations based on iGNSS QoS. Based on the results of several experiments using the developed methodology and algorithms, the following conclusions can be drawn:

- Point-based iGNSS QoS prediction can predict with high level of confidence, visible satellites at open sky locations. However, the confidence drops when predicting at obstructed locations. There are at least two reasons for this. One is that not all possible signal paths are captured by the simulation and another is that the high-sensitive GPS receivers used in the evaluation can lock onto weak satellite signals when operated in

obstructed environment. The models for visibility, availability, and accuracy can identify good iGNSS QoS for open sky locations (*Vis* > 0.8, *Av* =100%, and *Ac* > 0.8) and differentiate potentially poor iGNSS QoS with lower values of the prediction parameters.

- Overall, the results of the proposed methodology were highly accurate for the open sky route segments. However, differences between the prediction results and GPS baselines on the segments in obstructed areas indicted that on average, the methodology predicted relatively lower QoS than the GPS QoS baselines. Analyses of factors influencing the prediction results of the tracking-based prediction module and the segmentation module reveal the following:
    - Sampling distance affects computation time of the methodology and the chunk size produced by iGNSS QoS segmentation but it has insignificant impact on the results of tracking-based iGNSS QoS prediction.
    - Granularity of TINs affects computation time. It is recommended that, for driving mode, coarser TINs with objects higher than 20 m are used in visibility calculation (i.e., 2-zone LOS calculation) and for walking mode, finer TINs with objects near sidewalks are used in visibility calculation (i.e., 3-zone LOS calculation).
    - Time variation and surrounding environments affect iGNSS QoS on sidewalks more frequently and more significantly than roads.
- The satellite selection algorithm (MCSSA) can provide the subset of satellites with good geometry resulting in near-optimal DOP while requiring low computation time, which can help mitigating the computational burden caused by a large set of visible iGNSS satellites.
- The use of grid platforms can considerably improve performance of visibility calculations. Based on our analysis of the experimental results, the following conclusions can be drawn. First, the use of grid platforms can considerably improve performance of visibility calculations. Although the results do not support the real-time requirement of iGNSS QoS prediction, we shed some light on the possibilities of taking the grid computing approach for iGNSS QoS and navigation services. Second, the size of truncated LOS distance highly influences computation time of visibility calculation. Third, data distribution or replication is a determining factor in the grid environment. A

fully replicated data distribution provides shorter run times than a fully distributed data distribution. However, this method incurs considerable increase in storage overhead. This is not to say that storage is more valuable than time. Time is certainly considered more valuable than storage space in this application.

- Cloud computing is a viable approach for real-time geospatial applications, but our current feasibility study using GAE and GeoModel cannot efficiently handle data- and/or compute- intensive problems in iGNSS QoS prediction. The results of the TIN data retrieval experiment revealed that: (a) GAE performance decreases linearly with the number of point entities retrieved due to the entity search using geocells defined by GeoModel; (b) proximity query performs worse than bounding box query due to an iterative process of distance calculations; (c) search of triangle entities directly from GAE datastore using the entity key is relatively fast; and (d) GAE performance is unstable, especially for proximity query, due to the nature of resource sharing.

8.4 FUTURE RESEARCH

While iGNSS QoS prediction is new and promising, there are many questions and issues that need further investigation:

- *3D data representation*. Efficient handling of large scale TINs requires high-performance techniques and tools. In addition, TINs do not contain object properties such as materials (glass, concrete) and types (buildings, bridges, trees) and can cause shape distortion especially on boundary of objects. To improve the prediction performance and accuracy, other 3D data representations (e.g., virtual 3D city model), need to be investigated.
- *User dynamism*. This dissertation assumes that the user travels with a constant speed along a route when computing dynamic iGNSS QoS prediction. A more realistic model that takes into account user's speed and traffic information, among others, is needed in order to improve the prediction accuracy.
- *Segment sampling*. The current methodology uses a systematic sampling method with a fixed distance interval. If the sampling interval is too large, the prediction results may not

reflect the actual iGNSS QoS on segments, whereas if the sampling interval is too small, computation cost can considerably increase. Other sampling techniques (e.g., stratified sampling) that can adjust the sampling interval based on iGNSS QoS variations and geographic areas need to be investigated.

- *Validation of iGNSS QoS prediction with other constellations*. The current iGNSS QoS prediction methodology is evaluated with the GPS constellation only. The use of the combined GNSS constellations (e.g., GPS, GLONASS, Galileo) needs to be investigated.
- *Integration of iGNSS QoS prediction with the satellite selection algorithm (MCSSA)*. MCSSA has been tested separately from the current implementation of iGNSS QoS prediction. A strategy for incorporating MCSSA with the iGNSS QoS prediction methodology is needed in order to minimize computation cost while maximizing prediction accuracy.
- *Distributed algorithms for a grid-based iGNSS QoS simulation*. Using a grid computing environment is potentially a first step in achieving real-time iGNSS QoS prediction. However, new algorithms, in particular for visibility calculations at each sample point and QoS calculations from samples points along a route, are needed in order to efficiently exploit the benefits of grids.
- *Geoprocessing techniques and tools in clouds*. Although using a cloud (GAE with GeoModel) for distributed store and retrieval of TIN data is possible, GAE's performance, based on our current feasibility study, is unsuitable to meet the real-time requirement of iGNSS QoS prediction due to restrictions and limitations imposed by the current GAE version. For this reason, a more appropriate feasibility study must be performed and other cloud platforms for iGNSS QoS prediction will need to be investigated. Furthermore, utilizing cloud computing platforms that do not support geospatial applications requires development of geoprocessing techniques and tools designed specifically for cloud implementation and deployment, such as C2Geo (Karimi and Roongpiboonsopit, 2011); in particular APIs that allow developers to flexibly access and utilize cloud resources for geospatial applications are needed.

APPENDIX A

3D DATA

A.1 3D DATA SOURCE AND MODEL

The primary 3D data source of the Earth's surface used in simulation of iGNSS QoS prediction was LiDAR point cloud data. The study area was defined to cover several municipalities in Allegheny County, Pennsylvania, including the downtown area of Pittsburgh and the University of Pittsburgh's main campus. The LiDAR point cloud data was obtained from the Pennsylvania State Data Center. The LiDAR data for Allegheny County was collected as part of the PAMAP project in 2006 with a 1.4 m average point spacing (2 m maximum) and with a bare earth surface vertical accuracy of 18.5 cm RMSE (DCNR, n.d.). This data was in Log ASCII Standard (LAS) version 2.0 format, where each point was represented with a (x,y,z) coordinate. Data is in the tile form with the PAMAP 3.048-km (10,000-ft) tile index. A total of fifteen files were obtained and their extents are shown in Figure A.1. Each tile contains approximately 3 million points, at 90 megabytes (MB) of storage. High-resolution LiDAR points adequately represent the structures of both terrain and shapes of man-made objects. Figure A.2 shows the 3D representation of LiDAR points comparing to 3D models from Google Earth. Note that the 3D models of the three tall towers in the bottom right of Figure A.2(a) are missing due to unavailability of Google Earth's 3D models.

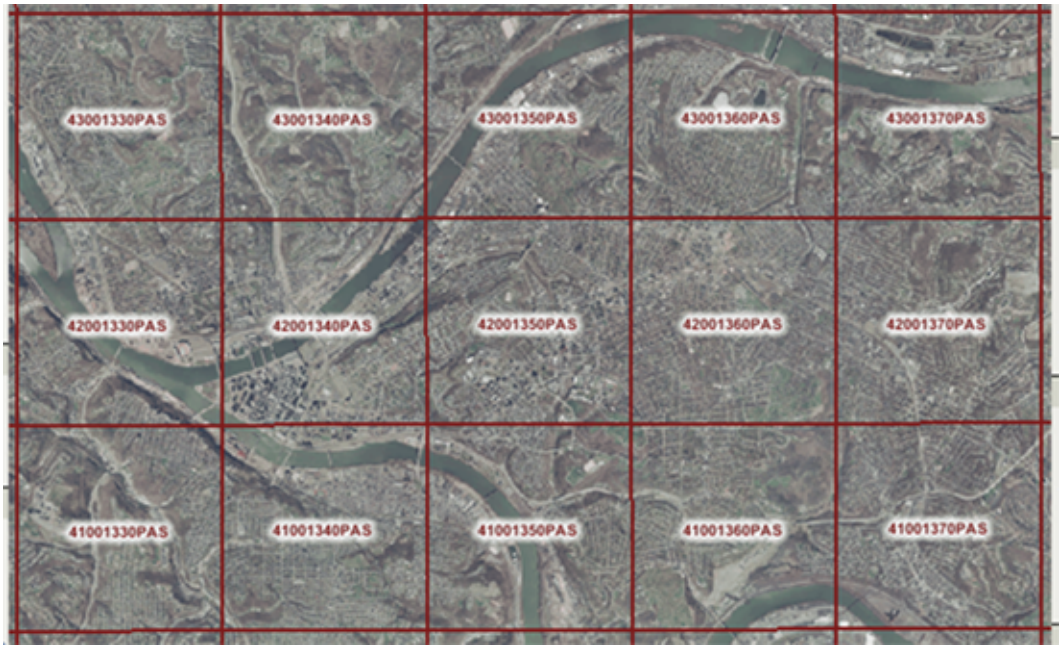
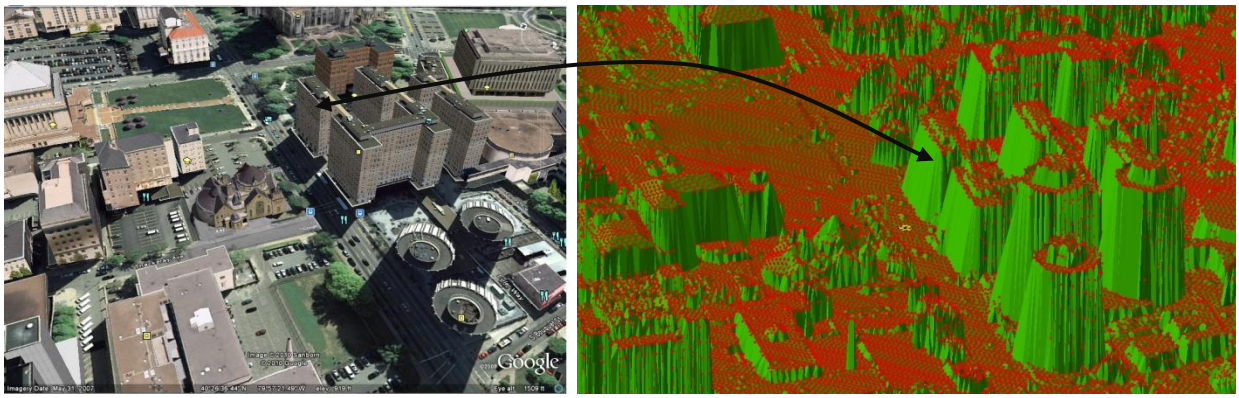


Figure A.1. Fifteen tiles for LiDAR point cloud



(a) Google Earth 3D (b) Point cloud data

Figure A.2. Google Earth's 3D map versus TIN constructed from point cloud data

To represent the structure of surfaces, the point cloud data was converted to a TIN model. The TIN model constructed from the point cloud contains a huge amount of nodes to represent the surface. Thus, it requires an efficient way to store, manage, and retrieve data. Although OGC has defined the data abstraction for TIN coverage to facilitate data exchange over the network,

currently there exists no GIS software packages or spatial-related tools with TIN data exchange capabilities.

We evaluated ArcGIS and Oracle Database 11g Spatial to accommodate TINs. ArcGIS provides 3D analysis tools to create and handle TINs but it has limited capabilities to efficiently handle large data files. In addition, as a proprietary product, ArcGIS provides limited choices of programming languages to developers. Oracle Database, on the other hand, has less GIS capabilities but can create, store, and manage large TINs efficiently. Although several proprietary data types for geographical data are defined in Oracle, it has adopted and implemented other geospatial data types defined by OGC and ISO standards, which support data interoperability. Standard TIN data types are expected to be implemented by Oracle in the near future. Oracle Database 11g (Release 1) was employed for simulation.

The Spatial extension of Oracle Database 11g (Release 1) provides a package, called SDO_TIN_PKG, for creating and querying TINs. The package uses the Delaunay triangulation as the default algorithm for generating a TIN. Oracle defines a data type, called SDO_TIN, to facilitate storing a TIN in its RDBMS. To store a TIN, two main tables are required: the base table for storing the SDO_TIN data and the block table for storing TIN blocks with several attributes (e.g., triangles, points, block id, block extent). For details of the SDO_TIN_PKG refer to (Kothuri et al., 2007). Figure A.3 demonstrates the steps taken to create TINs in Oracle. The LAStoOracle program is a java-based application, developed by Oracle, for reading the LAS file format and importing the LiDAR point cloud to Oracle.

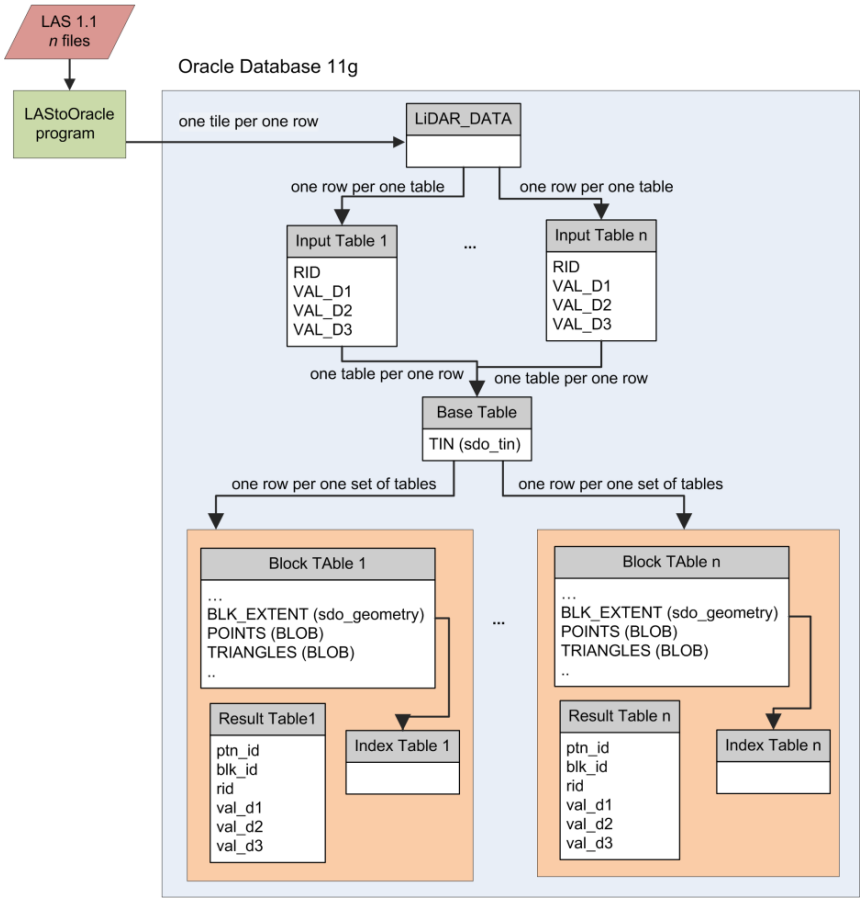


Figure A.3. Steps for constructing Oracle TINs from LiDAR files

A.2 TIN DATA LAYER PREPARATION

The steps for preparing and constructing the high-resolution TIN data layer used in Zone 1 LOS calculation was described in APPENDIX A.1; however, generating the high-elevation and bare-earth TIN data layers used in Zone 2 and Zone 3 LOS calculation, see Section 4.2.4.1, requires additional steps as demonstrated in Figure A.4.

The bare-earth TIN data layer was constructed from DEM. An example of DEM is shown in Figure 4.3(c). The original DEM was in raster format with 1-m resolution, which contains a large amount of grid data. To represent terrain using a TIN, sparse points representing changes in elevation are sufficient. This is based on the assumption that the geographic surface does not abruptly change in elevation. The reduced number of points used to create a TIN from DEM was

prepared by using a conversion tool (the Raster to TIN function) in ArcGIS. The z-tolerance parameter of the conversion tool was set at 3 m. The higher the z-tolerance, the smaller the number of points in TINs, resulting in a lower-resolution surface. Then, the TIN nodes generated by ArcGIS were imported into Oracle for constructing TIN tables.

The high-elevation TIN data layer, representing objects having a certain height above the ground, was prepared by subtracting the Digital Surface Model (DSM) generated from LiDAR points from the DEM. The result was the differences in elevation, which are the heights of objects above the ground. Only the cell having the difference more than the defined height was included in the TIN construction process. An example result of the elevation difference is shown in Figure 4.3(b). The points were later imported into Oracle for constructing TIN tables.

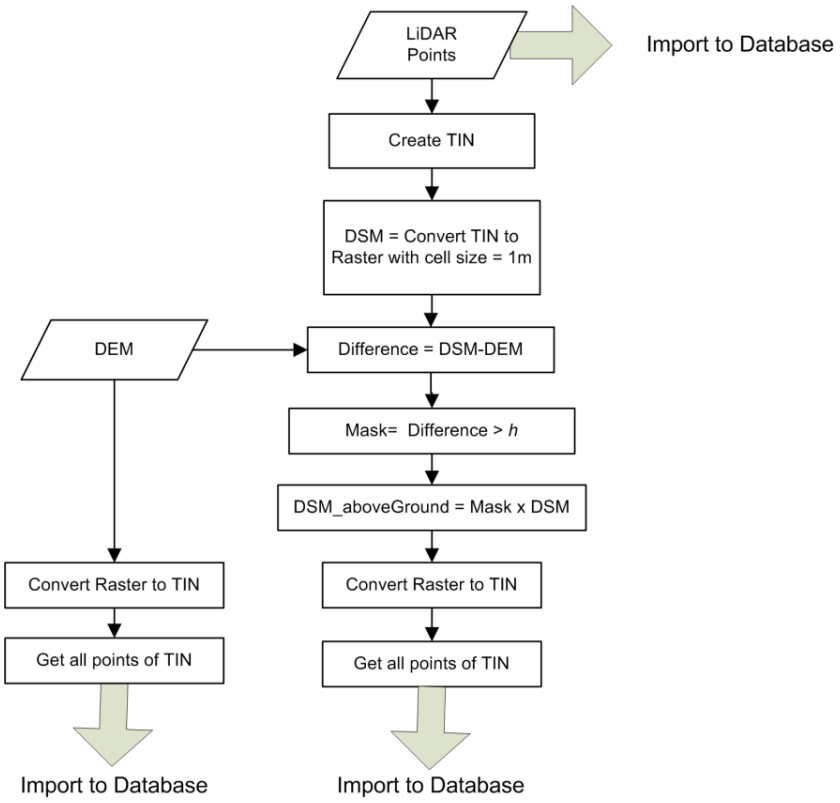


Figure A.4. Data preparation steps for TIN generation

APPENDIX B

EXAMPLES

B.1 EXAMPLE SCENARIO FOR MCSSA

To demonstrate how MCSSA works as described in Section 6.2.2, an example scenario for selecting six satellites is given. The user operates the iGNSS receiver using navigation information from GPS and Galileo. User's position is at 40.44° and longitude 79.96°. At a certain time, the receiver observes fourteen satellites (with elevation angle cutoff at 5°) consisting of seven GPS satellites and seven Galileo satellites. Table B.1 shows a list of LOS vectors and elevation angles for all visible satellites as well as the calculated parameters for each step (Steps1–4) of the algorithm.

The first four steps of the MCSSA are those in the MVA. In the first step, MCSSA selects satellite #224 as S_1 because of the largest elevation angle with respect to the receiver's position. In the second step, satellite #233 is selected as S_2 because of the angle between S_1 and S_2 ($\theta = 78.94°$) is closest to 109.5°. In the third step, satellite #9 is selected as S_3 because it gives the largest volume of the tetrahedron formed by S_1, S_2, S_3 and S_3'. In the fourth step, satellite #8 is selected as S_4 due to its maximum volume of the tetrahedron formed by S_1, S_2, S_3, and S_4.

To add the fifth and the sixth satellites, MCSSA determines the redundancy values. Table B.2 shows the redundancy values of all satellites in the remaining list for both steps. In the fifth step, satellite #20 is selected as S_5 because it is the least redundant to $S_{1\text{-}4}$ with the minimum *J*. In the final step, after MCSSA updates the remaining satellites list and the redundancy for each satellite with respect to S_1-S_5, satellite #231 which has the least redundancy value is selected as

S_6. Thus, the six satellites selected from the two GNSS constellations are satellites 8, 9, 20, 224, 231, and 233.

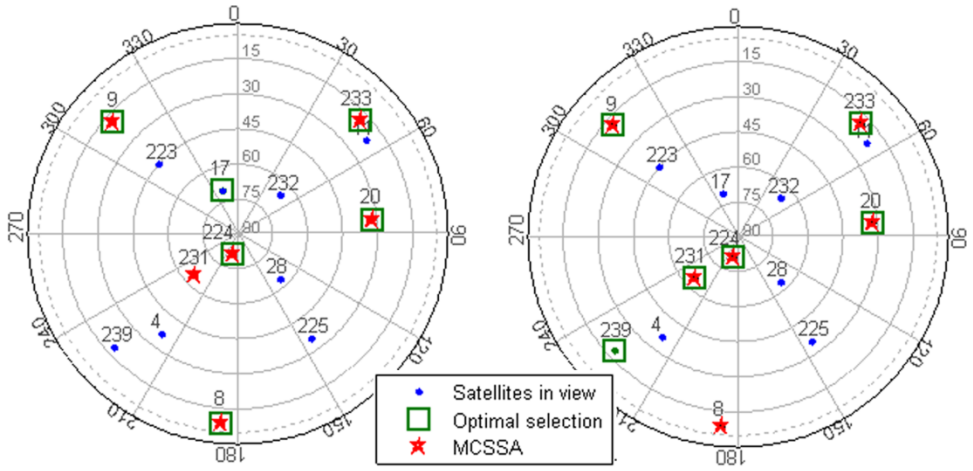
Based on the geometry of the satellites, the six satellites selected by MCSSA gives PDOP=1.694 and HDOP=1.325. The six satellites with the optimal PDOP=1.683 and HDOP=1.321 are also plotted in Figures B.1(a) and (b), respectively. To examine the result further, PDOPs of all 3003 combinations (selecting 6 satellites from 14 visible satellites) are calculated. Figure B.2(a) shows the plot of PDOPs for all combinations and the PDOP value of the satellite set selected by MCSSA. Figure B.2(b) shows similar details to Figure B.2(a) but using HDOP values. It is clear that MCSSA provides the set of satellites with the near optimal for both PDOP and HDOP.

Table B.1. Visible satellites for the example scenario

step 1						step 2	step 3	step 4
System	SV ID #	Elevation angle (degree)	*h* x	*h* y	*h* z	Angle to S_1 (degree)	V_3	V_4
GPS	4	36.48	-0.3272	-0.9395	-0.1019	45.56	0.1070	0.1494
	8	9.06	0.0419	-0.7621	-0.6461	72.25	0.2870	**0.2844**
	9	18.29	-0.7259	0.0466	0.6862	76.01	**0.2980**	-
	11	21.92	0.7216	0.2026	0.6620	74.92	0.0262	0.0040
	17	70.07	-0.0280	-0.5209	0.8531	27.64	0.0325	0.0650
	20	32.17	0.8892	-0.1946	0.4141	61.16	0.0981	0.0475
	28	63.51	0.4546	-0.8265	0.3319	22.77	0.0241	0.0457
Galileo	223	44.85	-0.4825	-0.3212	0.8149	49.72	0.1324	0.0459
	224	**81.22**	0.1123	-0.8417	0.5282	-	-	-
	225	35.63	0.6122	-0.7802	-0.1285	48.88	0.1316	0.1568
	231	64.48	-0.1611	-0.9171	0.3647	18.84	0.0176	0.0377
	232	64.98	0.3927	-0.4410	0.8070	32.69	0.0271	0.0532
	233	18.44	0.6558	0.2955	0.6947	**78.94**	-	-
	239	17.98	-0.5765	-0.7642	-0.2892	64.80	0.2002	0.2022

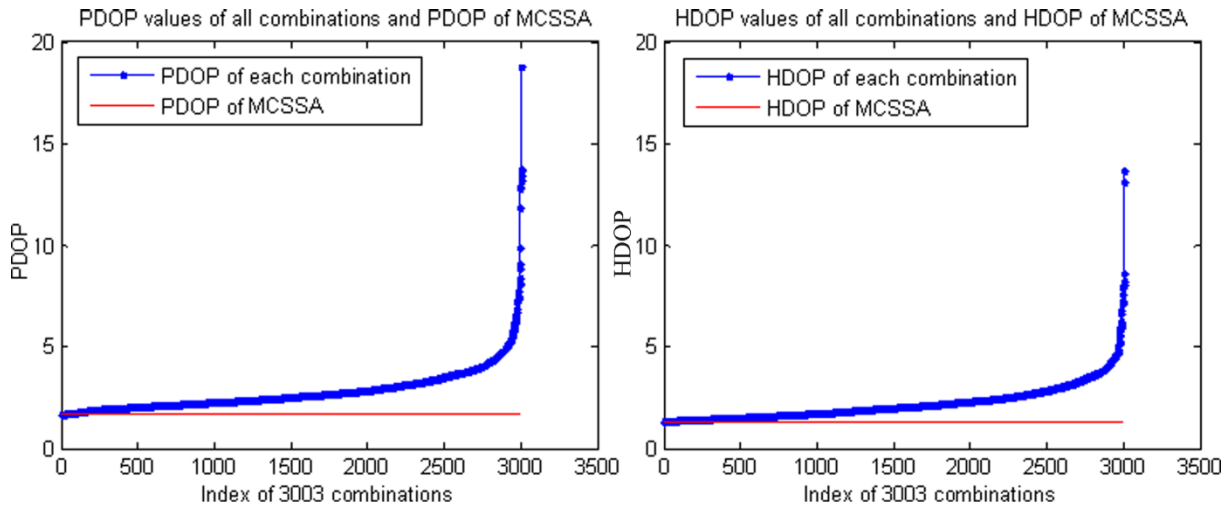
Table B.2. Remaining satellites after selecting four satellites with redundancy values

Satellite ID #	*J* of step 5	*J* of step 6
4	-1.1754	-2.1302
11	-1.2762	-0.7401
17	-1.3523	-1.9829
20	**-1.9238**	
28	-1.7206	-1.7336
223	-1.3336	-2.3319
225	-1.4854	-1.6586
231	-1.5509	**-2.4815**
232	-1.4571	-1.2737
239	-1.1638	-1.696



(a) PDOP (b) HDOP

Figure B.1. Skyplot of visible satellites for the example scenario representing the six selected satellites by MCSSA and the six satellites with (a) optimal PDOP and (b) optimal HDOP



(a) (b)

Figure B.2. All possible combinations and the MCSSA result using (a) PDOP and (b) HDOP

BIBLIOGRAPHY

AAP (2010, February 25). Drowned sailors GPS at least 100m out. Retrieved April 1, 2010

Al-Salami, A. (2009). TIN support in an open source spatial database. International Institute for Geo-information Science and Earth Observation (ITC), Enschede, The Netherlands.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., et al. (2009). Above the clouds: a Berkeley view of cloud computing (No. UCB/EECS-2009-28): EECS Department, University of California at Berkeley.

Beesley, B. (2002). Sky viewshed modeling for GPS use in the urban environment. University of South Carolina.

Berlich, R., Kunze, M. and Schwarz, K. (2005). Grid computing in Europe: from research to deployment. Australasian Workshop on Grid Computing and e-ResearchNewcastle, Australia, 21-27.

Berman, F., Fox, G. and Hey, A. (2003). *Grid computing: making the global infrastructure a reality*. John Wiley & Sons Inc.

Blower, J. (2010). GIS in the cloud: implementing a web map service on Google App Engine. 1st Intl. Conf. on Computing for Geospatial Research & Applications, June 21-23, Washington D. C..

Bowditch, N. (2002). *The American Practical Navigator*.

Bradbury, J. (2007). Prediction of urban GNSS availability and signal degradation using virtual reality city models. ION GNSS 20th International Technical Meeting of the Satellite Division, September 25-28, Fort Worth, TX.

Brauner, J., Foerster, T., Schaeffer, B. and Baranski, B. (2009). Towards a research agenda for geoprocessing services. The 12th AGILE International Conference on Geographic Information ScienceHanover, Germany.

Brunner, F. and Welsch, W. (1993). Effect of the troposphere on GPS measurements. *GPS World*, **4**, 42.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W., Wallach, D., Burrows, M., et al. (2008). Bigtable: a distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, **26**, 1-26.

Chen, Q. (2007). Airborne LiDAR data processing and information extraction. *Photogrammetric Engineering and Remote Sensing*, **73**, 109-112.

Constantinescu, A. and Landry Jr., R. (2005). GPS/Galileo/GLONASS hybrid satellite constellation simulator - GPS constellation validation and analysis. Proceedings of the Annual Meeting (Institute of Navigation), June 27-29, Cambridge, MA, 733-744.

Cornillon, P. (2009). Processing large volumes of satellite-derived sea surface temperature data - is cloud computing the way to go? Cloud Computing and Collaborative Technologies in the Geosciences Workshop, September 17-18, Indianapolis, IN.

D'Angelo, P., Fernández, A., Diez, J., Fossati, D., Maradi, L. and Gabaglio, V. (2005). Performance and visibility analysis for different Galileo/GPS receivers with the GRANADA Environment and Navigation Simulator. ENC/GNSS 2005 conference, July 19-22, Munich, Germany.

DCNR (n.d.). PAMAP program LiDAR. Retrieved February 16, 2010, from http://www.dcnr.state.pa.us/topogeo/pamap/lidar.aspx

Dennehy, K. (2010, February 10). Nokia's decision to offer free navigation: defensive or offensive? Retrieved February 24, 2010, from http://www.gpsworld.com/lbs/nokias-decision-offer-free-navigation-defensive-or-offensive-9496

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269-271.

Dolan, B. (2007, June 20). Navigation apps drive LBS success. *FierceDeveloper* Retrieved February 24, 2010, from http://www.fiercedeveloper.com/story/navigation-apps-drive-lbs-success/2007-06-20

Drane, C. and Rizos, C. (1998). *Positioning systems in intelligent transportation systems*. Artech House, Inc.

Driver, T. (2007). Long-term prediction of GPS accuracy: understanding the fundamentals. ION GNSS 20th International Technical Meeting of the Satellite Division, September 25-28, Fort Worth, TX.

Enright, A. J. and Ouzounis, C. A. (2000). GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics*, **16**, 451-457.

ESRI (2009, December 18). Spatial data service deployment utility for Windows Azure is available! Retrieved May 22, 2010, from http://blogs.esri.com/Dev/blogs/mapit/archive/2009/12/18/Spatial-Data-Service-Deployment-Utility-for-Windows-Azure-is-available_2100_.aspx

ESRI (2010). ArcGIS and the cloud. Retrieved June 7, 2010, from http://www.esri.com/technology-topics/cloud-gis/arcgis-and-the-cloud.html

European Space Agency (2007). GALILEO system simlation facility (GSSF). Retrieved 29 September, 2009, from http://www.gssf.info/

Feng, Y. (2003). Combined Galileo and GPS: a technical perspective. *Journal of Global Positioning Systems*, **2**, 67-72.

Filjar, R., Desic, S. and Huljenic, D. (2004). Satellite positioning for LBS: a Zagreb field positioning performance study. *Journal of Navigation*, **57**, 441-447.

Foster, I., Kesselman, C. and Tuecke, S. (2001). The anatomy of the grid: enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, **15**, 200.

Germroth, M. and Carstensen, L. (2005). GIS and satellite visibility: viewsheds from space. ESRI International User Conference 2005.

Ghemawat, S., Gobioff, H. and Leung, S. (2003). The Google file system. *ACM SIGOPS Operating Systems Review*, **37**, 29-43.

Grejner-Brzezinska, D. (2004). Telegeoinformatics: positioning and tracking approaches and technologies. In Karimi, H. A. and Hammad, A. (Eds.), *Telegeoinformatics: location-based computing and services*. Florida, US: CRC Press LLC.

Habrich, H. (1999). Geodetic applications of the global navigation satellite system (GLONASS) and of GLONASS/GPS combinations. University of Berne, Switzerland.

Hanjiang, X., Limin, T. and Long, S. (2008). A strategy to build a seamless multi-scale TIN-DEM database. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, July 3-11, Beijing, China.

Hannah, B. M. (2001). Modelling and simulation of GPS multipath propagation. Queensland University of Technology.

Hart, P. E., Nilsson, N. J. and B.Raphael. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybernetics*, **4**, 100-107.

Hein, G. W., Rodriguez, J. A. A., Wallner, S., Eissfeller, B., Pany, T. and Hartl, P. (2007). Envisioning a future GNSS system of systems: Part I. *InsideGNSS*, **2**, 58-62.

Hightower, J. and Borriello, G. (2001). A survey and taxonomy of location systems for ubiquitous computing. *IEEE Computer*, **34**, 57-66.

Hill, C. (2009). Experiences with atmosphere and ocean models on EC2. Cloud Computing and Collaborative Technologies in the Geosciences Workshop, September 17-18, Indianapolis, IN.

Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J. (2001). *GPS theory and practice* (5th ed.). Springer.

Hofmann-Wellenhof, B., Lichtenegger, H. and Wasle, E. (2008). *GNSS-global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer-Verlag Wien.

Jeannot, M., Boschetti, M., Godefroy, B., Leroy, S. and Martinez, F. (2005). Study of GNSS reception in mountainous area and benefits of surface correlation. The European Navigation Conference - GNSS 2005, July 19-22, Munich, Germany.

Jensen, A. B. O., Zabic, M., Overø, H. M., Ravn, B. and Nielsen, O. A. (2005). Availability of GNSS for road pricing in Copenhagen. Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation-ION GNSS, September, Long Beach, CA, 2951-2961.

Johnsonbaugh, R. and Schaefer, M. (2003). *Algorithms*. Prentice-Hall, Inc.

Kaplan, E. D. and Hegarty, C. J. (2006). *Understanding GPS: principles and applications* (2nd ed.). Artech house.

Karimi, H. (1996). Real-time optimal-route computation: a heuristic approach. *Journal of intelligent transportation systems,* **3**, 111-127.

Karimi, H., Liu, X., Liu, S. and Hammad, A. (2004). GPSLoc: framework for predicting Global Positioning System quality of service. *Journal of Computing in Civil Engineering,* **18**, 196-206.

Karimi, H., Zimmerman, B., Ozcelik, A. and Roongpiboonsopit, D. (2009). SoNavNet: a framework for social navigation networks. International Workshop on Location Based Social Networks (LBSN'09), November 3-6, Seattle, WA, 81-87.

Karimi, H. A., Conahan, T. and Roongpiboonsopit, D. (2006). A methodology for predicting performances of map-matching algorithms. 6th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2006), December 4-5, Hong Kong, 202-213.

Karimi, H. A. and Grejner-Brzezinska, D. A. (2004). GQMAP: Improving performance and productivity of mobile mapping systems through GPS quality of service. *Cartography and Geographic Information Science,* **31**, 167-177.

Karimi, H. A. and Roongpiboonsopit, D. (2011). C2GEO: Techniques and tools for real-time data-intensive geoprocessing in cloud computing. The 1st International Conference on Cloud Computing and Services Science, CLOSER 2011, May 7-9, Noordwijkerhout, The Netherlands.

Karimi, H. A., Roongpiboonsopit, D. and Kasemsuppakorn, P. (2011a). Uncertainty in personal navigation services. *Journal of Navigation,* **64**, 341-356.

Karimi, H. A., Roongpiboonsopit, D. and Wang, H. (2011b). Exploring real-time geoprocessing in cloud computing: navigation services case study. *Transactions in GIS*, **15**. (In press)

Karimi, H. A., Zimmerman, B., Nawn, D. and Sutovsky, P. (2010). Collaborative navigation systems for collision avoidance. 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), October 9-12, Chicago, IL, 1-9.

Karimi, H. A., Zimmerman, B., Roongpiboonsopit, D. and Rezgui, A. (2011c). Grid based geoprocessing for integrated global navigation satellite system simulation. *Journal of Computing in Civil Engineering*, **1**, 68.

Kasemsuppakorn, P. and Karimi, H. A. (2009a). Pedestrian network data collection through location-based social networks. The 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing, November 11-14, Washington D.C.

Kasemsuppakorn, P. and Karimi, H. A. (2009b). Personalised routing for wheelchair navigation. *Journal of Location Based Services*, **3**, 24-54.

Katsura, H., Miura, J., Hild, M. and Shirai, Y. (2003). A view-based outdoor navigation using object recognition robust to changes of weather and seasons. Proceedings IEEE/RSJ International Conference on Intelligent Robots and SystemsLas Vegas, NV, 2974-2979.

Kihara, M. and Okada, T. (1984). A satellite selection method and accuracy for the Global Positioning System. *Navigation*, **31**, 8-20.

Kim, K. S. and MacKenzie, D. (2009). Use of cloud computing in impact assessment of climate change. Free and Open Source Software for Geospatial (FOSS4GT), October 20-23, Sydney, Australia.

Kleijer, F., Odijk, D. and Verbree, E. (2008). Prediction of GNSS availability and accuracy in urban environments–case study schiphol airport. *Location Based Services and Telecartography: From Sensor Fusion to Ubiquitous* LBS, 387-406.

Kleusberg, A. (1990). Comparing GPS and GLONASS. *GPS World*, **1**, 52-54.

Kothuri, R., Godfrind, A. and Beinat, E. (2007). *Pro Oracle Spatial for Oracle Database 11g*. Apress.

Krause, A., Stoye, J. and Vingron, M. (2000). The SYSTERS protein sequence cluster set. *Nucleic acids research*, **28**, 270-272.

Kumar, P., Bapi, R. S. and Krishna, P. R. (2007). SeqPAM: a sequence clustering algorithm for Web personalization. *International Journal of Data Warehousing and Mining*, **3**, 29-53.

Langley, R. B. (1991). The mathematics of GPS. *GPS World*, **2**, 45-50.

Langley, R. B. (1999). Dilution of precision. *GPS World,* **10**, 52-59.

Lee, Y., Suh, Y. and Shibasaki, R. (2007). Ajax GIS application for GNSS availability simulation. *KSCE Journal of Civil Engineering,* **11**, 303-310.

Lee, Y., Suh, Y. and Shibasaki, R. (2008a). A GIS-based simulation to predict GPS availability along the Tehran Road in Seoul, Korea. *KSCE Journal of Civil Engineering,* **12**, 401-408.

Lee, Y., Suh, Y. and Shibasaki, R. (2008b). A simulation system for GNSS multipath mitigation using spatial statistical methods. *Computers and Geosciences,* **34**, 1597-1609.

Li, J., Ndili, A., Ward, L. and Buchman, S. (1999). GPS receiver satellite/antenna selection algorithm for the Stanford gravity probe B relativity mission. Institute of Navigation (ION) National Technical Meeting'Vision 2010: Present and Future, San Diego, CA, 541-550.

Li, J., Taylor, G., Kidner, D. and Ware, M. (2006). Prediction of GPS multipath effect using LiDAR digital surface models and building footprints. *Lecture notes in computer science,* **4295**, 42-53.

Liu, S. (2005). Geoprocessing Optimization in Grids. University of Pittsburgh.

Lohani, B. and Kumar, R. (2007). A model for GPS-GDOP prediction in urban environment using LiDAR data. 8th Conference on Optical 3-D Measurement Techniques, July 9-12, Zurich, Switzerland, 339-346.

Marais, J., Berbineau, M. and Heddebaut, M. (2005). Land mobile GNSS availability and multipath evaluation tool. *IEEE Transactions on Vehicular Technology*, **54**, 1697-1704.

Marais, J., Girard, P., Prestail, A., Franchineau, J. and de Verdalle, E. (2007). Galileo availability for urban buses. the 7th International Conference on ITS Telecommunications, June 6-8, Sophia Antipolis, 1-5.

Marais, J., Meunier, B. and Berbineau, M. (2000). Evaluation of GPS availability for train positioning along a railway line. IEEE Vehicular Technology Conference, September 24-28, Boston MA, 2060-2067.

Massatt, P. and Rudnick, K. (1991). Geometric formulas for dilution of precision calculations. *Navigation,* **37**, 379-391.

Mobile Marketing. (2011, May 27). Qualcomm annouces GLONASS support. Retrieved June 1, 2011, from http://m.mobilemarketingmagazine.com/mobilemarketing/i/article/qualcomm-annouces-glonass-support

Nurik, R. and Shen, S. (2009, December). Geospatial queries with Google App Engine using GeoModel. Retrieved September 22, 2010, from http://code.google.com/apis/maps/articles/geospatial.html#geomodel

O'Keefe, K., Ryan, S. and Lachapelle, G. (2002). Global availability and reliability assessment of the GPS and Galileo global navigation satellite systems. *Canadian Aeronautics and Space Journal*, **48**, 123-132.

Ochieng, W. Y., Quddus, M. A. and Noland, R. B. (2003). Map-matching in complex urban road networks. *Brazilian Journal of Cartography (Revista Brasileira de zartografia)*, **55**, 1–18.

Ochieng, W. Y., Sauer, K., Cross, P. A., Sheridan, K. F., Iliffe, J., Lannelongue, S., et al. (2001). Potential performance levels of a combined Galileo/GPS navigation system. *Journal of Navigation*, **54**, 185-197.

Ochieng, W. Y., Shardlow, P. J. and Johnston, G. (1999). Advanced Transport Telematics Positioning Requirements: An Assessment of GPS Performance in Greater London. *Journal of Navigation*, **52**, 342-355.

OfficialGoogleBlog. (2009, October 28). Announcing Google Maps Navigation for Android 2.0. Retrieved February 24, 2010, from http://googleblog.blogspot.com/2009/10/announcing-google-maps-navigation-for.html

Omnisdata. (2010). GIS Cloud beta: the next generation of GIS. Retrieved June 4, 2010, from http://www.giscloud.com/

Park, C. W. (2001). Precise relative navigation using augmented CDGPS. Ph.D. thesis, Stanford university, Stanford.

Park, S., Suresh, N. C. and Jeong, B. K. (2008). Sequence-based clustering for Web usage mining: A new experimental framework and ANN-enhanced K-means algorithm. *Data & Knowledge Engineering*, **65**, 512-543.

Parkinson, B. and Spilker, J. (1996). *The global positioning system: theory and applications* (Vol. 1). Aiaa.

Pemmaraju, S. and Skiena, S. (2003). "All-Pairs Shortest Paths" and "Transitive Closure and Reduction". In *Computational discrete mathematics: combinatorics and graph theory with Mathematica*. Cambridge, England: Cambridge University Press.

Poiker, T. K. (1990). The TIN Model. *NCGIA Core Curriculum in GIScience* Retrieved November 19, 2010, from http://www.ncgia.ucsb.edu/giscc/units/u056/u056.html

Porretta, M., Nepa, P., Manara, G. and Giannetti, F. (2008). Location, location, location. *Vehicular Technology Magazine*, **3**, 20-29.

Quddus, M. A. (2006). High integrity map matching algorithms for advanced transport telematics applications. Ph.D. thesis, Imperial College, London, United Kingdom.

Rainbow, J. and Clarke, P. (2002). Optimising navigational precision from potential GNSS constellations *The Hydrographic Journal*.

Ray, J. (2000). Mitigation of GPS code and carrier phase multipath effects using a multi-antenna system. Calgary, Alberta, Canada: University of Calgary.

Ray, J., Cannon, M. and Fenton, P. (1999). Mitigation of static carrier-phase multipath effects using multiple closely spaced antennas. *Jounal of The Institute of Navigation,* **46**, 193-202.

Reed, D. (2003). Grids, the TeraGrid and beyond. *Computer,* **36**, 62-68.

Rémy, B. and Moura, G. (2007). EnviNav: deterministic simulator of waves propagation in constrained environments for navigation and telecommunication. AGI User Exchange, August 29, Washington D.C.

Retscher, G. and Kealy, A. (2006). Ubiquitous positioning technologies for modern intelligent navigation systems. *Journal of Navigation,* **59**, 91-103.

Rizos, C. (2005). Trends in geopositioning for LBS, navigation and mapping. Proceedings International Symposium and Exhibition on Geoinformation, September 27-29, Penang, Malaysia.

Roche, K., Valdez, I. and Douglas, J. (2009). *Beginning Java Google App Engine* (1st ed.). Apress.

Rogers, S., Fiechter, C.-N. and Langley, P. (1999). An adaptive interactive agent for route advice. Proceedings of the third annual conference on Autonomous Agents, Seattle, Washington, United States, 198-205.

Rogerson, P. (2006). *Statistical methods for geography: a student guide*. Sage.

Roongpiboonsopit, D. and Karimi, H. (2010a). Comparative evaluation and analysis of online geocoding services. *International Journal of Geographical Information Science,* **24**, 1081-1100.

Roongpiboonsopit, D. and Karimi, H. (2010b). Quality assessment of online street and rooftop geocoding services. *Cartography and Geographic Information Science,* **37**, 301-318.

Roongpiboonsopit, D. and Karimi, H. A. (2009). A multi-constellations satellite selection algorithm for integrated GNSSs. *Journal of Intelligent Transportation Systems,* **13**, 127 – 141.

Sage, A. (2009). The real race in GNSS: To be the partner of choice with GPS. *GPS World,* **20**, 29.

Schäffer, B. and Baranski, B. (2009). Towards spatial related business processes in SDIs. 12th AGILE International Conference on Geographic Information Science, June, Hannover, Germany.

Seynat, C., Kealy, A. and Zhang, K. (2004). A performance analysis of future Global Navigation Satellite Systems. *Journal of Global Positioning Systems,* **3**, 232-241.

Skog, I. and Händel, P. (2009). In-car positioning and navigation technologies–a survey. *IEEE Transactions on Intelligent Transportation Systems,* **10**, 4-21.

Steed, A. (2004). Supporting mobile applications with real-time visualisation of GPS availability. *Lecture notes in computer science*, 373-377.

Stevens, R., Robinson, A. and Goble, C. (2003). myGrid: personalised bioinformatics on the information grid. *Bioinformatics,* **19**, 302-304.

Stoker, J., Greenlee, S., Gesch, D. and Menig, J. (2006). CLICK: The new USGS center for lidar information coordination and knowledge. *Photogrammetric Engineering and Remote Sensing,* **72**, 613-616.

Suh, Y. and Shibasaki, R. (2007). Evaluation of satellite-based navigation services in complex urban environments using a three-dimensional GIS. *IEICE Transactions on Communications,* **90**, 1816-1825.

Taylor, G., Li, J., Kidner, D., Brunsdon, C. and Ware, M. (2007). Modelling and prediction of GPS availability with digital photogrammetry and LiDAR. *International Journal of Geographical Information Science*, **21**, 1-20.

Taylor, G., Li, J., Kidner, D. and Ware, M. (2005). Surface modelling for GPS satellite visibility. The 5th International Workshop on Web and Wireless Geographical Information Systems, December 15-16, Lausanne, Switzerland, 281-295.

Toledo-Moreo, R., Santa, J., Ubeda, B., Piñana, C., Zamora-Izquierdo, M. A. and Gomez-Skarmeta, A. F. (2010). Performance aspects of navigation systems for GNSS-based road user charging. Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2010), September 21 - 24, Portland, Oregon, 1157 - 1165.

Toth, C. and Grejner-Brzezinska, D. (2004). Redefining the paradigm of modern mobile mapping: An automated high-precision road centerline mapping system. *Photogrammetric engineering and remote sensing,* **70**, 685-694.

USDoD. (2008). Global positioning system standard positioning service performance standard. Washington, DC: U.S. Department of Defense.

Verbree, E., Tiberius, C. and Vosselman, G. (2004). Combined GPS-Galileo positioning for location based services in urban environment. Proceedings of the Symposium on Location Based Services & Telecartography, January 28-29, Vienna, Austria, 99-107.

Verhagen, S. (2002). Studying the performance of global navigation satellite systems: a new software tool. *GPS World, 2010*, February 8.

Vrhovski, D., Moore, T. and Bennett, L. (2004a). GNSS-based road user charging. *The Journal of Navigation*, **57**, 1-13.

Vrhovski, D., Moore, T. and Bennett, L. (2004b). Simulating GPS in urban traffic environments. *GPS World*, **15**, 20-29.

Wang, Y., Wang, S. and Zhou, D. (2009). Retrieving and indexing spatial data in the cloud computing environment. *Lecture Notes in Computer Science, Cloud* Computing, 322-331.

Wormley, S. J. (2009, December 18). GPS errors & estimating your receiver's accuracy. Retrieved December 24, 2009, from http://edu-observatory.org/gps/gps_accuracy.html

Wu, F., Kubo, N. and Yasuda, A. (2003). A study of hybrid modernized GPS and Galileo Positionging in Japan. *Jounal of Japan Institute of Navigation,* **109**, 1-8.

Yang, C., Raskin, R., Goodchild, M. and Gahegan, M. (2010). Geospatial Cyberinfrastructure: Past, present and future. *Computers, Environment and Urban Systems,* **34**, 264-277.

Yarlagadda, R., Ali, I., Al-Dhahir, N. and Hershey, J. (2000). GPS GDOP metric. IEE Proceedings on Radar, Sonar and Navigation, 259-264.

Zahradnik, F. (2009). GPS-enabled phone market to grow 6.4 percent in 2009. Retrieved February 28, 2010, from http://gps.about.com/od/trends/a/gps_phone.htm

Zhang, K., Liu, G., Wu, F., Densley, L., Retscher, G., Vision, S., et al. (2008). An investigation of the signal performance of the current and future GNSS in typical urban canyons in Australia using a high fidelity 3D urban model. In Cartwright, W., Gartner, G., Meng, L. and Peterson, M. P. (Eds.), *Location Based Services and Telecartography II: From Sensor Fusion to Ubiquitous LBS* (pp. 407-420). Berlin Heidelberg: Springer.

Zhao, Y. (1997). *Vehicle location and navigation systems*. Artech House, Inc.

FREQUENTLY USED ACRONYMS

ANFIS :Adaptive Neural Fuzzy Interference System
ATT :Advanced Transport Telematics
DEM :Digital Elevation Model
DGPS :Differential Global Positioning System
DoD :Department of Defense
DOP :Dilution of Precision
DR :Dead Reckoning
DSM :Digital Surface Model
FIS :Fuzzy Inference System
FOC :Full Operation Capability
GAE :Google App Engine
GDOP :Geometric Dilution of Precision
GIS :Geographic Information System
GLONASS :GLObal NAvigation Satellite System
GNSS :Global Navigation Satellite System
GPS :Global Positioning System
GSSF :Galileo System Simulation Facility
HDOP :Horizontal Dilution of Precision
GST :GNSS Simulation Toolkit
HEA :Highest Elevation satellite selection Algorithm
iGNSS :integrated Global Navigation Satellite System
IGS :International GNSS Service
INS :Inertial Navigation System
ITS :Intelligent Transportation System

LBS :Location-Based Service

LiDAR :Light Detection And Ranging

LOS :Line-of-Sight

MCSSA :Multi-Constellation Satellite Selection Algorithm

MDR :Multipath-to-Direct Radio

MPI :Message Passing Interface

MVA :Maximum Volume Algorithm

NAVSTAR :NAVigation System with Timing And Ranging

NMEA :National Marine Electronics Association

OGSA-DAI :Open Source Grid Services Architecture for Data Access and Integration

OSSA :Optimal Satellite Selection Algorithm

PDOP :Position Dilution of Precision

PPS :Precise Positioning Service

PRN :Pseudorandom Noise

QOA :Quasi-Optimal satellite selection Algorithm

QoS :Quality of Service

RFID :Radio Frequency Identification

RMSE :Root Mean Square Errors

RNP :Required Navigation Performance

SMA :Sequence Merging Algorithm

SNR :Signal to Noise Ratio

SPS :Standard Positioning Service

TDOP :Time Dilution of Precision

TIN :Triangulated Irregular Network

TOA :Time of Arrival

TTFF :Time-to-First Fix

UERE :User Equivalent Range Error

UTC :Coordinated Universal Time

UWB :Ultra Wide-Band

VDOP :Vertical Dilution of Precision

VPS :Vision-based Positioning System

VRML :Virtual Reality Modeling Language

WLAN :Wireless Local Area Network

WMS :Web Map Service

WPS :Wi-Fi Positioning System